Discrete least squares approximation with polynomial vectors

Marc Van Barel
Adhemar Bultheel

Report TW 190, May 1993

Department of Computer Science, K.U. Leuven

Abstract

We give the solution of a discrete least squares approximation problem in terms of orthonormal polynomial vectors. The degrees of the polynomial elements of these vectors can be different. An algorithm is constructed computing the coefficients of recurrence relations for the orthonormal polynomial vectors. In case the function values are prescribed in points on the real axis or on the unit circle, variants of the original algorithm can be designed which are an order of magnitude more efficient. Although the recurrence relations require all previous vectors to compute the next orthonormal polynomial vector, in the real or the unit-circle case only a fixed number of previous vectors are required.

Keywords : orthogonal polynomial vectors, discrete least squares.
AMS(MOS) Classification : Primary : 41A20, Secondary : 65F25, 65D05, 65D15, 30E10.

Discrete least squares approximation with polynomial vectors

Marc Van Barel and Adhemar Bultheel*

May 18, 1993

Abstract

We give the solution of a discrete least squares approximation problem in terms of orthonormal polynomial vectors. The degrees of the polynomial elements of these vectors can be different. An algorithm is constructed computing the coefficients of recurrence relations for the orthonormal polynomial vectors. In case the function values are prescribed in points on the real axis or on the unit circle, variants of the original algorithm can be designed which are an order of magnitude more efficient. Although the recurrence relations require all previous vectors to compute the next orthonormal polynomial vector, in the real or the unit-circle case only a fixed number of previous vectors are required.

1 Introduction

In this paper, we want to solve the following discrete least squares approximation problem: Given the points $z_i \in \mathbb{C}$, $i = 1, 2, \ldots, m$ and the weight vectors $F_i \in \mathbb{C}^{1\times n}$, compute the polynomial vector $P \in \mathbb{C}[z]^{n\times 1}$ with a componentwise upper bound for the degree

$$\partial P \leq \Delta := [\delta_1, \ldots, \delta_n] \quad \text{(componentwise)}, \qquad \Delta \in (\mathbb{N} \cup \{-1\})^{n\times 1},$$

such that

$$\sum_{i=1}^{m} P^H(z_i) F_i^H F_i P(z_i)$$

is minimal.

The problem in this form will always have the trivial solution $P \equiv 0$. Therefore, we add the condition that one of the elements of P has to be monic, i.e. we have precise degree for the monic component. We will solve this discrete least squares approximation problem using polynomial vectors orthogonal with respect to the discrete inner product

$$\langle P, Q\rangle := \sum_{i=1}^{m} P^H(z_i) F_i^H F_i Q(z_i).$$

We will give an algorithm to compute the building blocks of a recurrence relation from which these orthogonal polynomial vectors can be computed. We show that if all the points z_i are real or all the points z_i are on the unit circle, the complexity of the algorithm can be reduced an order of magnitude.

In previous publications [4, 14, 15], we have considered special cases of the approximation problem described above. In [14], we gave an algorithm to solve the problem with real points z_i, $n = 2$ and $\delta_1 = \delta_2$. The algorithm is a generalization of the algorithm of Reichel [11], which looks for the optimal polynomial fitting some given function values in some real

*Department of Computing Science, K.U.Leuven, Celestijnenlaan 200 A, 3001 Heverlee

points z_i in a least squares sense. Reichel's algorithm itself is based on the Rutishauser-Gragg-Harrod algorithm [13, 10, 1] for the computation of Jacobi matrices. Similar results were obtained in [3, 7]. In section 10, we investigate the real point case for arbitrary n and arbitrary degrees $\delta_i, i = 1, 2, \ldots, n$.

Based on the inverse unitary QR algorithm for computing unitary Hessenberg matrices [2], Reichel, Ammar and Gragg [12] solve the approximation problem when the given function values are taken in points on the unit circle. In [15], we generalized this for $n = 1$ to $n = 2$ with equal degrees $\delta_1 = \delta_2$. Section 11 handles the general problem on the unit circle. When $n = 2$, we refer the reader to [4], which summarizes [14] and [15] and handles the case of arbitrary degrees δ_1 and δ_2.

In [14, 15], we have given some numerical examples showing that the algorithms can be used to compute rational interpolants or rational approximants in a linearized discrete least squares sense. In section 8, we give the conditions for having an interpolating polynomial vector. In a next publication, we shall show how we can use the theory developed here, to compute matrix rational interpolants or matrix rational approximants in a linearized discrete least squares sense.

2 Discrete least squares approximation problem

Definition 2.1 (inner product, norm) *Given the points $z_i \in \mathbb{C}$, (not necessarily different from each other) and the weight vectors $F_i \in \mathbb{C}^{1\times n}$, $i = 1, 2, \ldots, m$ we consider the following discrete inner product $\langle P, Q\rangle$ for two polynomial vectors $P, Q \in \mathbb{C}[z]^{n\times 1}$:*

$$\langle P, Q\rangle := \sum_{i=1}^{m} P^H(z_i) F_i^H F_i Q(z_i). \tag{1}$$

The norm $\|P\|$ of a polynomial vector $P \in \mathbb{C}[z]^{n\times 1}$ is defined as:

$$\|P\| := \sqrt{\langle P, P\rangle}.$$

We consider the following approximation problem

Definition 2.2 (discrete least squares approximation problem) *Given the points $z_i \in \mathbb{C}$ and the weight vectors $F_i \in \mathbb{C}^{1\times n}$, $i = 1, 2, \ldots, m$, the degree vector $\Delta := [\delta_1, \delta_2, \ldots, \delta_m]^T \in (\mathbb{N} \cup \{-1\})^{n\times 1}$ and some degree index $\nu_\Delta \in \{1, 2, \ldots, n\}$.*
With $\bar{\Delta} := (\Delta, \nu_\Delta)$ (the extended degree vector) and $P := [P_1, P_2, \ldots, P_n]^T \in \mathbb{C}[z]^{n\times 1}$, consider the sets $\mathcal{P}_\Delta$ and $\mathcal{P}_{\bar{\Delta}}$

$$\begin{aligned} \mathcal{P}_\Delta &:= \{P \in \mathbb{C}[z]^{n\times 1} \mid \partial P \leq \Delta\}, \\ \mathcal{P}_{\bar{\Delta}} &:= \{P \in \mathcal{P}_\Delta \mid \partial P_{\nu_\Delta} = \delta_{\nu_\Delta} \text{ and } P_{\nu_\Delta} \text{ is monic}\}. \end{aligned}$$

The discrete least squares approximation problem looks for the polynomial vector P such that $\|P\| = \min_{Q\in\mathcal{P}_{\bar{\Delta}}} \|Q\|$.

3 Orthonormal polynomial vectors

To solve the discrete least squares approximation problem, we could easily transform it into a linear algebra problem. Note that $F_i P(z_i) \in \mathbb{C}$ is a scalar. Therefore, the original problem is equivalent to the m linear equations

$$F_i P(z_i) = 0, \quad i = 1, 2, \ldots, m$$

which have to be solved in a least squares sense, i.e.

$$\sum_{i=1}^{m} |r_i|^2 \text{ is minimal with } r_i = F_i P(z_i)$$

(with $P \in \mathcal{P}_{\bar{\Delta}}$). Because $\mathcal{P}_\Delta$ is a $\mathbb{C}$-vector space having dimension $|\Delta| := \sum_{i=1}^{n}(\delta_i + 1)$, we can choose a basis for $\mathcal{P}_\Delta$ and write out the least squares problem using coordinates with respect to this basis. Introducing the normality condition, i.e. P_{ν_Δ} has to be monic, we can eliminate one of the coordinates. We obtain an $m \times (|\Delta| - 1)$ least squares problem. The amount of computational work is proportional to $m|\Delta|^2$ (e.g. using the normal equations or the QR factorization).

Assume however that we have an orthonormal basis for $\mathcal{P}_\Delta$ such that the basis vectors $B_j := [B_{j,1}, B_{j,2}, \ldots, B_{j,n}]$ satisfy $\partial B_{j,\nu_\Delta} < \delta_{\nu_\Delta}$, $j = 1, 2, \ldots, |\Delta| - 1$, and $\partial B_{|\Delta|,\nu_\Delta} = \delta_{\nu_\Delta}$, then we can write every $P \in \mathcal{P}_\Delta$ in a unique way as:

$$P = \sum_{j=1}^{|\Delta|} B_j a_j, \quad a_j \in \mathbb{C}.$$

Because P_{ν_Δ} has to be monic of degree δ_{ν_Δ}, $a_{|\Delta|}$ is fixed. The other coordinates a_j, $j = 1, 2, \ldots, |\Delta| - 1$ can be choosen freely. We get

$$\begin{aligned} \|P\|^2 &= \langle P, P \rangle \\ &= \left\langle \sum_{j=1}^{|\Delta|} B_j a_j, \sum_{j=1}^{|\Delta|} B_j a_j \right\rangle \\ &= \sum_{j=1}^{|\Delta|} |a_j|^2 \quad (\text{because } \langle B_i, B_j \rangle = \delta_{ij}). \end{aligned}$$

Therefore, to minimize $\|P\|$, we can put a_j, $j = 1, 2, \ldots, |\Delta| - 1$ equal to zero or

$$P = B_{|\Delta|} a_{|\Delta|} \quad \text{and} \quad \|P\| = |a_{|\Delta|}|.$$

Hence, to solve the least squares approximation problem we can compute the orthonormal polynomial vector $B_{|\Delta|}$ and this will give us the solution (up to a scalar multiplication to make it monic).

Up till now, the degree condition for the orthonormal basis vectors B_j was very mild. There is still a lot of freedom in choosing the B_j. We could start with any basis for $\mathcal{P}_{\Delta'}$ with $\Delta' := \Delta - U_{\nu_\Delta}$ and $U_j := [0, 0, \ldots, 0, 1, 0, \ldots, 0]^T$ (1 on the j-th place).

Using the inner product, we can apply Gram-Schmidt to get an orthonormal basis for $\mathcal{P}_{\Delta'}$. Finally we can add an arbitrary polynomial vector $\in \mathcal{P}_{\bar{\Delta}}$ and use Gram-Schmidt to obtain an orthonormal basis for $\mathcal{P}_\Delta$. Let us concentrate for a while on the scalar case ($n = 1$). Suppose that we are not only interested in the solution of the least squares problem with a fixed degree (vector) Δ but also in the solutions of degree $\Delta^{(k)} := \Delta + k$, $k = 1, 2, \ldots$. If we have an orthonormal basis $\mathbb{B}^{(k)}$ for $\Delta^{(k)}$, it is easy to get an orthonormal basis for $\Delta^{(k+1)}$. Because $\Delta^{(k+1)'} := \Delta^{(k+1)} - 1 = \Delta^{(k)}$, we can use the orthonormal basis of the previous step and use Gram-Schmidt to orthogonalize an arbitrary polynomial of degree $\Delta^{(k+1)}$ with respect to these orthonormal basis vectors.

If we start initially with $\Delta = 0$, we get a sequence of orthonormal basis vectors $\{B_0, B_1, B_2, \ldots\}$ with corresponding degree sequence $\{0, 1, 2, \ldots\}$. The only possibility for the degree index $\nu_{\Delta^{(k)}}$ in the scalar case is: $\nu_{\Delta^{(k)}} = 1$. However, we need this degree index in the vector case ($n > 1$). This generates more freedom in the choice of the degree vector sequence corresponding to the sequence of orthonormal polynomial vectors.

Suppose we are not only interested in a solution having a fixed extended degree vector $\bar{\Delta} := (\Delta, \nu_\Delta)$ but also in the solutions with

$$\bar{\Delta}^{(k)} := (\Delta^{(k)}, \nu_{\Delta^{(k)}})$$

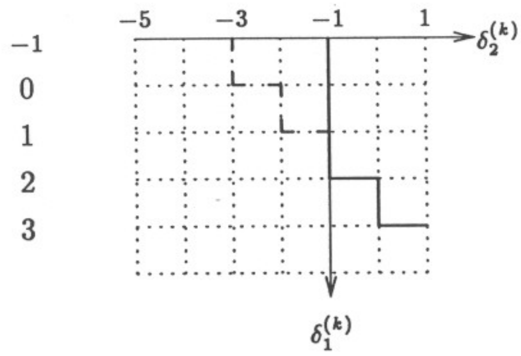

Figure 3.1: The original and the projected polyline of example 3.1

where

$$\begin{cases} k =: np + q \quad (0 < q \leq n) \quad (p, q \in \mathbb{Z}) \\ \Delta^{(k)} := \Delta - U^1_{\nu_\Delta} + (p+1)U - U^0_q \\ \nu_{\Delta^{(k)}} = q \\ U^1_j := [\underbrace{1, 1, \ldots, 1}_{j \text{ ones}}, 0, \ldots, 0]^T \\ U^0_j := [\underbrace{0, 0, \ldots, 0}_{j \text{ zeros}}, 1, \ldots, 1]^T \\ U := [1, 1, \ldots, 1, 1]^T. \end{cases}$$

We get for $k = \nu_\Delta$, i.e. $p = 0$ and $q = \nu_\Delta$, that $\Delta^{(k)} = \Delta$, hence that the solution has extended degree $\bar{\Delta} = \bar{\Delta}^{(\nu_\Delta)} = (\Delta, \nu_\Delta)$.
If we put the degree vectors in an n-dimensional table, we want to walk along a "diagonal". In the scalar case, it is clear that we can start our sequence of degrees by taking $\Delta = 0$. Then, $\mathcal{P}_\Delta$ has dimension one. For the vector case, things are a little bit more complicated. The idea is to arrive in $\Delta^{(k)} = \Delta \in \mathbb{Z}^n$ for some k. We shall follow a diagonal path in the n-dimensional space $\mathbb{Z}^{n\times 1}$. Each move on the diagonal from $\Delta^{(k)}$ to $\Delta^{(k)} + U$ will be decomposed in n elementary steps in each of the coordinate directions $\Delta^{(k+1)} = \Delta^{(k)} + U^1_1$, $\Delta^{(k+2)} = \Delta^{(k)} + U^1_2$, …, $\Delta^{(k+n)} = \Delta^{(k)} + U^1_n$ which results in a staircase-like polyline. This works quite well as soon as $\Delta^{(k)} \geq 0$ (componentwise). The starting point of this diagonal however will be outside the positive part of the coordinate system. As soon as some $\delta_i < 0$, the corresponding polynomial will be zero and it will remain zero, no matter how negative δ_i will get. This means that whenever $\Delta^{(k)}$ falls outside $(\mathbb{N} \cup \{-1\})^{n\times 1}$, $\mathcal{P}_{\Delta^{(k)}}$ will be equal to some $\mathcal{P}_{\Delta^{(l)}}$ with $\Delta^{(l)} \in (\mathbb{N} \cup \{-1\})^{n\times 1}$. Therefore, we shall project the polyline onto the part $(\mathbb{N} \cup \{-1\})^{n\times 1}$ of $\mathbb{Z}^{n\times 1}$, such that $\Delta^{(k)} < \Delta^{(k+1)}$ for all $k \geq 0$, which means that $\dim \mathcal{P}_{\Delta^{(k+1)}} = \dim \mathcal{P}_{\Delta^{(k)}} + 1$, starting with $\Delta^{(0)} = -U$, which corresponds to $\mathcal{P}_{\Delta^{(0)}} = \{[0, 0, \ldots, 0]^T\}$ with $\dim \mathcal{P}_{\Delta^{(0)}} = 0$.

Example 3.1 Take $n = 2$, $\Delta = [3, 1]^T$ and $\nu_\Delta = 2$. Figure 3.1 shows the original and projected polylines in $\mathbb{Z}^2$. The same results are summarized in the following table

	polyline in $\mathbb{Z}^{n\times 1}$												
k	…	-6	-5	-4	-3	-2	-1	0	1	2	3	4	…
p	…	-4	-3	-3	-2	-2	-1	-1	0	0	1	1	…
q	…	2	1	2	1	2	1	2	1	2	1	2	…
$\Delta^{(k)}$	…	-1	0	0	1	1	2	2	3	3	4	4	…
	…	-3	-3	-2	-2	-1	-1	0	0	1	1	2	…
$\nu_{\Delta^{(k)}}$	…	2	1	2	1	2	1	2	1	2	1	2	…
	projected polyline in $(\mathbb{N}\cup\{-1\})^{n\times 1}$ (renumbering by algorithm 3.1)												
k		0	1		2		3	4	5	6	7	8	…
$\Delta^{(k)}$		-1	0		1		2	2	3	3	4	4	…
		-1	-1		-1		-1	0	0	1	1	2	…
$\nu_{\Delta^{(k)}}$			1		1		1	2	1	2	1	2	…

Since all components of $\Delta^{(k)}, k \leq -6$ are negative, $\mathcal{P}_{\Delta^{(k)}} = \{[0,0]^T\}$ for all $k \leq -6$. Furthermore, we have $\mathcal{P}_{\Delta^{(-5)}} = \mathcal{P}_{\Delta^{(-4)}}$ and $\mathcal{P}_{\Delta^{(-3)}} = \mathcal{P}_{\Delta^{(-2)}}$. ◇

We will therefore change the numbering and select only those $\Delta^{(k)}$ such that

$$\dim \mathcal{P}_{\Delta^{(k+1)}} = \dim \mathcal{P}_{\Delta^{(k)}} + 1.$$

Therefore we introduce shift parameters defined by

$$\Delta^\star := [\delta_1^\star, \ldots, \delta_n^\star]^T := \Delta - U^1_{\nu_\Delta}.$$

For notational simplicity, we assume that

$$\delta_1^\star \geq \delta_2^\star \geq \ldots \geq \delta_n^\star \geq 0$$

which can always be obtained after reordering.

The reader can check that the following algorithm generates a sequence of degree indices $\nu_{\Delta^{(k)}}$ and degree vectors $\Delta^{(k)}$ corresponding to the projected polyline such that

$$\dim \mathcal{P}_{\Delta^{(k+1)}} = \dim \mathcal{P}_{\Delta^{(k)}} + 1, \quad k \geq 0.$$

Algorithm 3.1 *Given the shift parameters*

$$\delta_1^\star \geq \delta_2^\star \geq, \ldots, \delta_n^\star \geq 0, \quad \delta_i^\star \in \mathbb{N},$$

we construct the two sequences

$$\{\Delta^{(k)}\} \quad \textit{and} \quad \{\nu_{\Delta^{(k)}}\}$$

$k \leftarrow 0$
for $\delta := -\delta_1^\star, -\delta_1^\star + 1, \ldots$ **do**
  for $i := 1, 2, \ldots, n$ **do**
    if $\delta + \delta_i^\star \geq 0$ **then**

$$\begin{cases} k \leftarrow k+1 \\ \Delta^{(k)} \leftarrow \max\{\delta U + \Delta^\star - U_i^0, -U\}, \quad \textit{(componentwise)} \\ \nu_{\Delta^{(k)}} \leftarrow i \end{cases}$$

with

$$\begin{aligned} U &:= [1,1,\ldots,1]^T \\ \Delta^\star &:= [\delta_1^\star, \delta_2^\star, \ldots, \delta_n^\star]^T := \Delta - U^1_{\nu_\Delta} \\ U_j^1 &:= [\underbrace{1,1,\ldots,1}_{j\ ones},0,0,\ldots,0]^T \\ U_i^0 &:= [\underbrace{0,0,\ldots,0}_{i\ zeros},1,1,\ldots,1]^T. \end{aligned}$$

k	$\Delta^{(k)}$	$\nu_{\Delta^{(k)}}$
1	$(0,-1,-1,\ldots,-1)$	1
2	$(1,-1,-1,\ldots,-1)$	1
$\vdots$	$\vdots$	$\vdots$
$\delta_1^\star-\delta_2^\star+1$	$(\delta_1^\star-\delta_2^\star,-1,-1,\ldots,-1)$	1
$\delta_1^\star-\delta_2^\star+2$	$(\delta_1^\star-\delta_2^\star,0,-1,\ldots,-1)$	2
$\delta_1^\star-\delta_2^\star+3$	$(\delta_1^\star-\delta_2^\star+1,0,-1,\ldots,-1)$	1
$\delta_1^\star-\delta_2^\star+4$	$(\delta_1^\star-\delta_2^\star+1,1,-1,\ldots,-1)$	2
$\vdots$	$\vdots$	$\vdots$
	$(\delta_1^\star-\delta_3^\star,\delta_2^\star-\delta_3^\star-1,-1,\ldots,-1)$	1
	$(\delta_1^\star-\delta_3^\star,\delta_2^\star-\delta_3^\star,-1,\ldots,-1)$	2
$\delta_1^\star+\delta_2^\star-2\delta_3^\star+3$	$(\delta_1^\star-\delta_3^\star,\delta_2^\star-\delta_3^\star,0,-1,-1)$	3
$\vdots$	$\vdots$	$\vdots$
$\sum_{k=1}^{j-1}\delta_k^\star-(j-1)\delta_j^\star+j$	$(\delta_1^\star-\delta_j^\star,\delta_2^\star-\delta_j^\star,\ldots,\delta_{j-1}^\star-\delta_j^\star,0,-1,\ldots,-1)$	j
$\vdots$	$\vdots$	$\vdots$

Table 3.1: The sequences $\{\Delta^{(k)}\}_{k=1}^{\infty}$ and $\{\nu_{\Delta^{(k)}}\}_{k=1}^{\infty}$

	$-\delta_1^\star$		$\ldots$		$-\delta_2^\star$				$-\delta_3^\star$				$-\delta_n^\star$			$\delta \rightarrow$
1	$\circledast$	$\star$	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	
2	0	0	$\ldots$	0	$\circledast$	$\star$	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	
3	0	0	$\ldots$	0	0	0	$\ldots$	0	$\circledast$	$\star$	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	
4	0	0	$\ldots$	0	0	0	$\ldots$	0	0	0	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		
$n-1$	0	0	$\ldots$	0	0	0	$\ldots$	0	0	0	$\ldots$	$\star$	$\star$	$\star$	$\ldots$	
n	0	0	$\ldots$	0	0	0	$\ldots$	0	0	0	$\ldots$	0	$\circledast$	$\star$	$\ldots$	
$i \downarrow$																

Table 3.2: Degree structure table

Algorithm 3.1 generates the sequences $\{\Delta^{(k)}\}_{k=1}^{\infty}$ and $\{\nu_{\Delta^{(k)}}\}_{k=1}^{\infty}$ given in table 3.1. Another way to represent this result is as follows. Based on the shift parameters $\delta_1^\star \geq \delta_2^\star \geq \ldots \geq \delta_n^\star \geq 0$, we can construct the degree structure table 3.2. Algorithm 3.1 runs through the entries of this table column by column, starting at the top of each column going downwards. A $\star$ indicates that $\delta U + \Delta^\star - U_i^0$ is a degree vector which we have not encountered before. A 0 indicates that $\delta U + \Delta^\star - U_i^0$ gives no new degree structure. The entries where a new component comes into play are indicated as $\circledast$.

In the next section, we will introduce several indices to facilitate the formulation of the algorithm described in section 5.

4 Indices

In the next section, we give an algorithm with input our initial data (the points z_i, the weights F_i) and output the building blocks of a recurrence relation generating the desired orthonormal polynomial vectors. This transformation process is influenced by the shift parameters

$$\Delta^\star := [\delta_1^\star, \delta_2^\star, \ldots, \delta_n^\star]^T$$

with

$$\delta_1^\star \geq \delta_2^\star \geq \ldots \geq \delta_n^\star \geq 0, \quad \delta_i \in \mathbb{N}.$$

Because of this ordering, we shall start with $\mathcal{P}_{\Delta^{(0)}} = [0, 0, \ldots, 0]^T$. Then the first component will become nonzero. Its degree will raise till $\delta_1^\star - \delta_2^\star$. The next step is to introduce a second component which is nonzero, following a staircase in the subspace $[\delta_1, \delta_2, -1, \ldots, -1]^T$, until you reach $[\delta_1^\star - \delta_3^\star, \delta_2^\star - \delta_3^\star, -1, \ldots, -1]^T$. Now we introduce a nonzero polynomial in the third position, etc. The indices k of $\Delta^{(k)}$ where the nonzero polynomial in position j is introduced will be denoted by μ_j. We call them jump indices. So, the n shift parameters $\delta_1^\star, \ldots, \delta_n^\star$ will define uniquely the n jump indices

$$\mu_1 = 1 < \mu_2 < \mu_3 < \ldots < \mu_n \tag{2}$$

with

$$\mu_j = \sum_{k=1}^{j-1} \delta_k^\star - (j-1)\delta_j^\star + j, \quad j = 1, 2, \ldots, n.$$

For a proof we refer to theorem 7.3.

Once we have the jump indices, we define the so-called pivot indices π_k, $k = 1, 2, \ldots, m$ as follows ($\mu_{n+1} = +\infty$)

if $k = \mu_j$ then $\pi_k = j$
else $\mu_j < k < \mu_{j+1}$ for some $j \in \{1, 2, \ldots, n\}$ and then $\pi_k = k - j + n$

Why we define the pivot indices in this way will become clear later on when we shall show that the algorithm which we describe below will indeed give the solution of our problem having the prescribed degree structure. The previous definition of the pivot indices means that we basically take the sequence of numbers $\{n+1, n+2, n+3, \ldots, m-n\}$ and introduce the numbers $j = 1, 2, \ldots, n$ just before position μ_j, where we renumber each time.

Example 4.1 Take $n = 3$ with jump indices $\mu_1 = 1$, $\mu_2 = 4$, $\mu_3 = 5$. We start with the sequence $\{4, 5, 6, 7, 8, 9, 10, 11, \ldots, m-3\}$

1 is introduced before the position μ_1: $\{1, 4, 5, 6, 7, 8, 9, 10, 11, \ldots, m-2\}$,
2 is introduced before the position μ_2: $\{1, 4, 5, 2, 6, 7, 8, 9, 10, 11, \ldots, m-1\}$,
3 is introduced before the position μ_3: $\{1, 4, 5, 2, 3, 6, 7, 8, 9, 10, 11, \ldots, m\}$.

◇

The pivot indices play an important role in the definition of the algorithm. They define the position (i, π_i) of the pivot elements in the elimination procedure which we describe below.

5 The algorithm

The algorithm starts with a scheme which looks as follows:

$$\begin{array}{c|cccc} F_1 & z_1 & & & \\ F_2 & & z_2 & & \\ \vdots & & & \ddots & \\ F_m & & & & z_m \end{array} \quad =: \quad \begin{array}{c|c} F & \Lambda \end{array}$$

and transforms this using similarity transformations on Λ into

$$[Q^H F \mid Q^H \Lambda Q] = Q^H [F \mid \Lambda] \begin{bmatrix} I_n & \\ & Q \end{bmatrix}$$

(Q unitary) such that $[Q^H F \mid Q^H \Lambda Q]$ has zeros below the pivot positions (i, π_i), $i = 1, 2, \ldots, m$.
The following algorithm will do the job:

Algorithm 5.1 *Transformation of the initial data matrix* $D := [F \mid \Lambda]$ *into a matrix* $[Q^H F \mid Q^H \Lambda Q]$ *having zeros below the pivot elements*

for $i := 1$ **to** m **do**
  for $j := 1$ **to** $i-1$ **do**
    * *make element* d_{i,π_j} *zero*
    by using a Givens rotation (or reflection) J^H
    with the pivot element (j, π_j)*:*
      $D \leftarrow J^H D$
    * $D \leftarrow D \begin{bmatrix} I_n & \\ & J \end{bmatrix}$ *(similarity transformation)*

Algorithm 5.1 constructs

$$\sum_{i=1}^{m}(i-1) = (m-1)m/2$$

Givens rotations. For a certain i and j, the Givens rotation is applied to the left on 2 vectors of length $(i+n+1-j)$ and to the right on 2 vectors of length $\leq (j+n+1)$. The total number of Givens rotations applied to 2 numbers is therefore limited by

$$\begin{aligned} & \sum_{i=1}^{m}\sum_{j=1}^{i-1}[(i+n+1-j)+(j+n+1)] \\ = \; & \frac{m(m+1)(3m+1)}{6} + (2n+2-1)\frac{m(m+1)}{2} - (2n+2)m \\ = \; & O(m^3/2). \end{aligned}$$

Counting 4 multiplications for each application of a Givens rotation, this results in $O(2m^3)$ multiplications. Note that also a Householder variant of Algorithm 5.1 could be designed.

6 Recurrence relations for the columns of the unitary transformation matrix Q

In the previous section we have transformed the initial data matrix $D := [F \mid \Lambda]$ into

$$Q^H[F \mid \Lambda]\begin{bmatrix} I_n & 0 \\ 0 & Q \end{bmatrix} =: [E \mid G].$$

We can write

$$F = QE, \tag{3}$$
$$\Lambda Q = QG. \tag{4}$$

Knowing $E =: [e_{i,j}]$ and $G =: [g_{i,j}]$, we can reconstruct the columns Q_k of Q, $k = 1, 2, 3, \ldots, m$ based on the pivot indices. There are the following two possibilities:

a) $1 \leq \pi_k \leq n$: We know that $e_{i,\pi_k} = 0$, $i > k$, because E is zero below pivot position (k, π_k). Therefore, writing out equality (3) for the π_k-th column gives us (F'_j denotes the j-th column of F)

$$F'_{\pi_k} = [Q_1 Q_2 \ldots Q_k]\begin{bmatrix} E'_{\pi_k} \end{bmatrix}$$

with

$$E_{\pi_k} = \begin{bmatrix} E'_{\pi_k} \\ 0 \end{bmatrix}.$$

So, we can write Q_k as

$$e_{k,\pi_k} Q_k = F'_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k} Q_i. \tag{5}$$

b) $\pi_k - n =: \pi'_k > 0$. We know that $g_{i,\pi'_k} = 0$, $i > k$.
Writing out equality (4) for the π'_k-th column gives us:

$$\Lambda Q_{\pi'_k} = [Q_1 Q_2 \dots Q_k] \begin{bmatrix} G'_{\pi'_k} \end{bmatrix}$$

with

$$G_{\pi'_k} = \begin{bmatrix} G'_{\pi'_k} \\ 0 \end{bmatrix}.$$

So, we can write Q_k based on the previous columns of Q as:

$$g_{k,\pi'_k} Q_k = \Lambda Q_{\pi'_k} - \sum_{i=1}^{k-1} g_{i,\pi'_k} Q_i. \tag{6}$$

Note that $k > \pi'_k$ because $1 \le \tau_k \le n$, $k > 1$, with

$$\tau_k := k - \pi'_k = \#\{\pi_j | 1 \le \pi_j \le n, j < k\}.$$

As long as e_{k,π_k} and g_{k,π'_k} are different from zero, we can use (5) and (6) as a recurrence relation to compute the columns Q_k, $k = 1, 2, 3, 4, \dots$.

7 Recurrence relations for a sequence of orthonormal polynomial vectors

Similar to the recurrence relations (5) and (6) for the columns Q_k, we can construct a sequence of polynomial vectors $\{\phi_k\}_{k=1}^{\infty}$, $\phi_k \in \mathbb{C}[z]^{n\times 1}$ as follows:

a) $1 \le \pi_k \le n$:

$$e_{k,\pi_k} \phi_k(z) = U_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k} \phi_i(z) \tag{7}$$

b) $\pi_k - n =: \pi'_k > 0$:

$$g_{k,\pi'_k} \phi_k(z) = z\phi_{\pi'_k}(z) - \sum_{i=1}^{k-1} g_{i,\pi'_k} \phi_i(z). \tag{8}$$

Theorem 7.1 (relationship between Q_k and $\phi_k(z)$) *Let F_k denote the rows of F and F'_k the columns of F:*

$$[F'_1, F'_2, \dots, F'_n] := F =: \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}.$$

Then

$$Q_k = F^\star \phi_k^\star$$

with

$$F^\star := \textit{block diagonal}\,\{F_1, F_2, \dots, F_m\}, \quad \textit{and} \quad \phi_k^\star := \begin{bmatrix} \phi_k(z_1) \\ \vdots \\ \phi_k(z_m) \end{bmatrix}.$$

Proof. True for $k = 1$, because ($U_1 = [1, 0, \ldots, 0]^T$)

$$e_{1,1}Q_1 = F'_1 = F^\star \begin{bmatrix} U_1 \\ U_1 \\ \vdots \\ U_1 \end{bmatrix},$$

$$e_{1,1}\phi_1(z) = U_1, \quad \text{hence} \quad e_{1,1}\phi_1^\star = \begin{bmatrix} U_1 \\ U_1 \\ \vdots \\ U_1 \end{bmatrix}.$$

Thus

$$Q_1 = F^\star \phi_1^\star.$$

Induction hypothesis: Suppose the theorem is true for Q_i, $i = 1, 2, \ldots, k-1$.

a) $1 \leq \pi_k \leq n$: Take the recurrence relation (5) for Q_k:

$$e_{k,\pi_k} Q_k = F'_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k} Q_i.$$

We use the induction hypothesis:

$$Q_i = F^\star \phi_i^\star, \quad i = 1, 2, \ldots, k-1,$$

to get

$$\begin{aligned} e_{k,\pi_k} Q_k &= F^\star \begin{bmatrix} U_{\pi_k} \\ U_{\pi_k} \\ \vdots \\ U_{\pi_k} \end{bmatrix} - F^\star \sum_{i=1}^{k-1} e_{i,\pi_k} \phi_i^\star \\ &= F^\star (e_{k,\pi_k} \phi_k^\star). \end{aligned}$$

b) $\pi_k - n > 0$: The proof is similar.

□

Using the connection between the polynomial vectors $\phi_k(z)$ and the columns Q_k of the unitary transformation matrix Q, we get

Theorem 7.2 (orthonormality of ϕ_k) *The polynomial vectors, defined by (7) and (8), satisfy*

$$\langle \phi_k, \phi_l \rangle = \delta_{kl}$$

where the inner product is defined in (1).

Proof. This follows from the orthogonality of the columns Q_k:

$$\begin{aligned} \langle \phi_k, \phi_l \rangle &= \sum_{i=1}^{m} \phi_k(z_i)^H F_i^H F_i \phi_l(z_i) \\ &= Q_k^H Q_l \\ &= \delta_{kl} \end{aligned}$$

□

At this point, we have given an algorithm to compute the recurrence coefficients for a sequence of orthonormal polynomial vectors ϕ_k. Now, we want to show that our choice of the jump indices $\mu_1, \mu_2, \ldots, \mu_n$ as defined in (2) gives indeed the desired degree structure of the orthonormal polynomial vectors ϕ_k which was proposed by algorithm 3.1.

Theorem 7.3 *If we couple the jump indices μ_j to the shift parameter $\delta_j^\star$ in the following way:*

$$\begin{aligned} \mu_1 &= 1 \\ \mu_j &= \sum_{k=1}^{j-1} \delta_k^\star - (j-1)\delta_j^\star + j, \quad j = 2,3,\ldots,n \end{aligned}$$

the orthonormal polynomial vectors ϕ_k, have the degree structure given by algorithm 3.1.

Proof. We have the following relationship between the jump indices $\{\mu_j\}_{j=1}^n$, the pivot indices $\{\pi_k\}_{k=1}^\infty$ and the degree structure of ϕ_k (using recurrence relations (7) and (8)):

k	π_k	degree structure of ϕ_k	$\nu_{\Delta(k)}$
$\mu_1 = 1$	1	$(0,-1,-1,\ldots,-1)$	1
μ_1+1	$n+1$	$(1,-1,-1,\ldots,-1)$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
μ_2-1	$n+\mu_2-2$	$(\delta_1^\star-\delta_2^\star,-1,-1,\ldots,-1)$	1
μ_2	2	$(\delta_1^\star-\delta_2^\star,0,-1,\ldots,-1)$	2
μ_2+1	$n+\mu_2-1$	$(\delta_1^\star-\delta_2^\star+1,0,-1,\ldots,-1)$	1
μ_2+2	$n+\mu_2$	$(\delta_1^\star-\delta_2^\star+1,1,-1,\ldots,-1)$	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
μ_3-1	$n+\mu_3-3$	$(\delta_1^\star-\delta_3^\star,\delta_2^\star-\delta_3^\star,-1,\ldots,-1)$	2
μ_3	3	$(\delta_1^\star-\delta_3^\star,\delta_2^\star-\delta_3^\star,0,-1,-1)$	3
μ_3+1	$n+\mu_3-2$	$(\delta_1^\star-\delta_3^\star+1,\delta_2^\star-\delta_3^\star,0,-1,-1)$	1
μ_3+2	$n+\mu_3-1$	$(\delta_1^\star-\delta_3^\star+1,\delta_2^\star-\delta_3^\star+1,0,-1,-1)$	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
μ_n-1	μ_n	$(\delta_1^\star-\delta_n^\star,\delta_2^\star-\delta_n^\star,\ldots,\delta_{n-1}^\star-\delta_n^\star-1,-1)$	$n-1$
μ_n	n	$(\delta_1^\star-\delta_n^\star,\delta_2^\star-\delta_n^\star,\ldots,\delta_{n-1}^\star-\delta_n^\star,0)$	n
μ_n+1	μ_n+1	$(\delta_1^\star-\delta_n^\star+1,\delta_2^\star-\delta_n^\star,\ldots,\delta_{n-1}^\star-\delta_n^\star,0)$	1
μ_n+2	μ_n+2	$(\delta_1^\star-\delta_n^\star+1,\delta_2^\star-\delta_n^\star+1,\ldots,\delta_{n-1}^\star-\delta_n^\star,0)$	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Note that

$$(\mu_j - 1) - \mu_{j-1} = (\delta_{j-1}^\star - \delta_j^\star)(j-1)$$

is a multiple of $(j-1)$.
We can also look at table 3.2 to get the same result. The degree structure of an orthonormal polynomial vector computed by using recurrence relation (7) is indicated as ⊛. The other entries $\star$ are coupled to recurrence relation (8). Note that for an entry in the table with

$$-\delta_j^\star < \delta \le -\delta_{j+1}^\star \text{ and } 1 \le i \le j,$$

the value of π in (8) is such that ϕ_π is coupled to entry $(i, \delta - 1)$, i.e. lying on the same row (strict degree for the same i) but with all degrees decreased by one. □

Note that if we want to use the orthonormal polynomial vectors ϕ_k to solve the discrete least squares approximation problem of definition 2.2, we only have to compute $\phi_{|\Delta|}$ using the recurrence relations (7) and (8). Therefore, algorithm 5.1 can be adapted only computing those entries of E and G needed in the recurrence relations. The computational work will then be proportional to $m|\Delta|^2$ instead of m^3.

8 Singular case

Until now, we assumed all the entries e_{k,π_k} and g_{k,π'_k} to be different from zero, which we call the regular case.
Suppose now that these can be zero. Suppose that the first entry equal to zero is

(a) e_{k,π_k} : In this case, we can not use recurrence relation (7) to compute $\phi_k(z)$. However, we can compute a polynomial vector ϕ'_k as follows:

$$\phi'_k(z) = U_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k}\phi_i(z)$$

From (5), we know that

$$\begin{aligned} 0 = e_{k,\pi_k}Q_k &= F'_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k}Q_i \\ &= F'_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k}F^\star\phi_i^\star \\ &= F^\star \begin{bmatrix} U_{\pi_k} \\ U_{\pi_k} \\ \vdots \\ U_{\pi_k} \end{bmatrix} - \sum_{i=1}^{k-1} e_{i,\pi_k}F^\star\phi_i^\star \\ &= F^\star\phi_k'^\star \end{aligned}$$

Hence, $F_j\phi'_k(z_j) = 0$, $j = 1, 2, \ldots, m$.

(b) g_{k,π'_k}: As in (a), we can prove that

$$\phi'_k(z) = z\phi_\pi(z) - \sum_{i=1}^{k-1} g_{i,\pi'_k}\phi_i(z)$$

satisfies

$$F_j\phi'_k(z_j) = 0, \quad j = 1, 2, \ldots, m.$$

In the regular case, the least squares approximation error $||P|| = |a_{|\Delta|}|$ is different from zero. In the singular case, this error is zero and ϕ'_k is an interpolating polynomial vector for the given data.

Note that in the regular case the inner product from definition 2.1 is a true inner product for the $\mathbb{C}$-vector space $\mathcal{P}_{\Delta(k)}$, i.e. as long as e_{j,π_j} and g_{j,π'_j} are different from zero.

9 Related orthonormal polynomial vectors and matrices

We can consider orthonormal polynomial vectors with respect to the generalized inner product

$$\langle P, Q\rangle := \sum_{k'=1}^{m'} P(z_{k'})^H \begin{bmatrix} F_{k'}^{(1)} \\ \vdots \\ F_{k'}^{(l)} \end{bmatrix}^H \begin{bmatrix} F_{k'}^{(1)} \\ \vdots \\ F_{k'}^{(l)} \end{bmatrix} Q(z_{k'})$$

with

$$P, Q \in \mathbb{C}[z]^{n\times 1}$$

and

$$F_{k'}^{(j)} \in \mathbb{C}^{1\times n}, \quad k' = 1, 2, \ldots, m', \quad j = 1, 2, \ldots, l.$$

This inner product can be written as

$$\langle P, Q\rangle = \sum_{k'=1}^{m'} \sum_{j=1}^{l} P(z_{k'})^H F_{k'}^{(l)^H} F_{k'}^{(l)} Q(z_{k'})$$

which can always be rewritten as:

$$\langle P, Q\rangle = \sum_{k=1}^{m} P(z_k)^H F_k^H F_k Q(z_k)$$

reducing the problem of constructing a corresponding sequence of orthonormal polynomial vectors to the original problem.
To get orthonormal polynomial matrices, we consider the following inner product:

$$\langle P, Q\rangle := \sum_{k=1}^{m} P(z_k)^H F_k^H F_k Q(z_k) \in \mathbb{C}^{l\times l} \tag{9}$$

with $P, Q \in \mathbb{C}[z]^{n\times l}$.
Taking the shift parameters $\Delta^\star$, we can easily represent all polynomial matrices having a degree

$$\leq [\delta_1 U + \Delta^\star - U^0_{j_1}, \ldots, \delta_l U + \Delta^\star - U^0_{j_l}]$$

using the orthonormal polynomial vectors $\phi_k(z)$.
By grouping together l of these orthonormal polynomial vectors, we get (a kind of) orthonormal polynomial matrices with respect to (9).
We get the "classical" orthonormal polynomial matrices by setting $l = n$,

$$\Delta^\star := [\delta_1^\star, \delta_2^\star, \ldots, \delta_n^\star]^T = 0$$

and taking members of $\{\phi_k\}_{k=1}^{\infty}$ in groups of n columns to form a sequence of orthonormal polynomial $(n\times n)$-matrices. For more details, see e.g. [5, 8, 9, 6]

10 Recurrence relations if all points z_i are real

If $z_i \in \mathbb{R}$, $i = 1, 2, \ldots, m$, then $G := Q^H \Lambda Q$ is Hermitian because

$$G^H = (Q^H \Lambda Q)^H = Q^H \Lambda Q = G.$$

Because $g_{k,j} = 0$, $j < \pi_k$, also $g_{j,k} = 0$, $j < \pi_k$.
The recurrence relation (7) to compute a sequence of orthonormal polynomial vectors will not change, but recurrence relation (8) will have a smaller number of terms in the right-hand side:

$$g_{k,\pi'_k}\phi_k(z) = z\phi_{\pi'_k}(z) - \sum_{i=\lambda_k}^{k-1} g_{i,\pi'_k}\phi_i(z). \tag{10}$$

with

$$\lambda_k := \pi'_{\pi'_k} := \pi_{\pi'_k} - n = k - \tau_k - \tau_{\pi'_k}.$$

The number η_k of polynomial vectors ϕ_i in the right-hand side of (10) is equal to:

$$\begin{aligned}\eta_k &= (k-1) - \lambda_k + 1 = k - \lambda_k \\ &= (k - \pi'_k) + (\pi'_k - \pi'_{\pi'_k}) \\ &= \tau_k + \tau_{\pi'_k} \leq 2\tau_k \leq 2n.\end{aligned}$$

Hence, to compute ϕ_k we need not more than the previous $2n$ orthonormal polynomial vectors ϕ_i while in the general case we have to use all the previous ϕ_i. Let us look at some special cases of this result:

a) When $n = 1$ (the scalar case), the recurrence relation (10) is just the classical 3-term recurrence relation for scalar orthonormal polynomials:

$$g_{k,k-1}\phi_k(z) = (z - g_{k-1,k-1})\phi_{k-1}(z) - g_{k-2,k-1}\phi_{k-2}(z), \quad k > 1$$

with

$$e_{1,1}\phi_1(z) = U_1 \quad \text{and} \quad \phi_0(z) \equiv 0.$$

b) When $\pi_i = i$, $i = 1, 2, \ldots, n$, we use recurrence relation (7) to compute $\phi_1, \phi_2, \ldots, \phi_n$. For $k > n$, recurrence relation (10) gives us:

$$g_{k,k-n}\phi_k(z) = z\phi_{k-n}(z) - \sum_{i=k-2n}^{k-1} g_{i,k-n}\phi_i(z)$$

with $\phi_i \equiv 0$, $i < 1$.

The computational work of algorithm 5.1 reduces by an order of magnitude in case all z_i are real. Each Givens rotation (or reflection) involves vectors of length $\leq 2(n+1)$ instead of vectors of length $i+n+1-j$. Applying the Givens rotation to the left, requires $\leq 8(n+1)$ multiplications. Applying the Givens rotation to the right only requires 8 multiplications because of symmetry considerations. Therefore, the total number of multiplications is limited by

$$\begin{aligned} \sum_{i=1}^{m}\sum_{j=1}^{i-1}[8(n+1)+8] &= 4(n+2)m(m-1) \\ &= O(4(n+2)m^2) \end{aligned}$$

which is an order of magnitude m smaller compared to the general case. If we are only interested in $\phi_{|\Delta|}$, the computational work is proportional to $m|\Delta|$.

We can transform the recurrence relation for the polynomial vectors ϕ_k into a block 3-term recurrence relation. Because of the notational complexity, we give only an example showing this equivalence.

Example 10.1 Suppose the transformed data matrix has the following structure:

$$[E\ \ G] = \left[\begin{array}{ccc|ccccccc}
\circledast & \times & \times & \times & \times & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & \circledast & \times & \times & \times & 0 & 0 & 0 \\
0 & \circledast & \times & 0 & \times & \times & \times & \times & 0 & 0 \\
0 & 0 & \times & 0 & \circledast & \times & \times & \times & \times & \times \\
0 & 0 & \times & 0 & 0 & \circledast & \times & \times & \times & \times \\
0 & 0 & \circledast & 0 & 0 & 0 & \times & \times & \times & \times \\
0 & 0 & 0 & 0 & 0 & 0 & \circledast & \times & \times & \times
\end{array}\right]$$

$$=: \left[\begin{array}{ccc|cccccc}
\times & \times & \times & \times & C_0 & & 0 & 0 & 0 \\
0 & \times & \times & \times & A_0 & C_1' & 0 & 0 & 0 \\
0 & \times & \times & 0 & & & \times & 0 & 0 \\
0 & 0 & D_0 & 0 & B_0 & & \times & \times & \times \\
0 & 0 & & 0 & & A_1' & \times & \times & \times \\
0 & 0 & \times & 0 & 0\quad 0 & & \times & \times & \times \\
0 & 0 & 0 & 0 & 0\quad 0 & B_1' & \times & \times & \times
\end{array}\right].$$

The pivot elements (k, π_k), $k = 1, 2, \ldots, 7$ are indicated by $\circledast$. If we define

$$\begin{aligned} \Phi_{-1}(z) &:= 0_3, \quad \Phi_0(z) := I_3 = [U_1\ \ U_2\ \ U_3], \\ \Phi_1(z) &:= [\phi_1(z)\ \ U_2 - e_{1,2}\phi_1(z)\ \ U_3 - e_{1,3}\phi_1(z)], \end{aligned}$$

$$\begin{aligned}
\Phi_2(z) &:= [\phi_2(z)\ \ U_2 - \sum_{i=1}^{2} e_{i,2}\phi_i(z)\ \ U_3 - \sum_{i=1}^{2} e_{i,3}\phi_i(z)], \\
\Phi_3(z) &:= [\phi_2(z)\ \ \phi_3(z)\ \ U_3 - \sum_{i=1}^{3} e_{i,3}\phi_i(z)], \\
\Phi_4(z) &:= [\phi_4(z)\ \ \phi_5(z)\ \ U_3 - \sum_{i=1}^{5} e_{i,3}\phi_i(z)], \\
\Phi_5 &:= [\phi_4(z)\ \ \phi_5(z)\ \ \phi_6(z)], \quad \Phi_6(z) := [\phi_7(z)],
\end{aligned}$$

they satisfy the block 3-term recurrence relation

$$\Phi_k(z) = \Phi_{k-1}(z)\beta_{k-1} + \Phi_{k-2}(z)\alpha_{k-1}, \quad k = 1, 2, \ldots, 6,$$

with

$$\begin{aligned}
\beta_0 &:= \begin{bmatrix} \frac{1}{e_{1,1}} & -\frac{e_{1,2}}{e_{1,1}} & -\frac{e_{1,3}}{e_{1,1}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \alpha_0 := 0_3 \\
\beta_1 &:= \begin{bmatrix} \frac{(z-g_{1,1})}{g_{2,1}} & -\frac{(z-g_{1,1})}{g_{2,1}e_{2,2}} & -\frac{(z-g_{1,1})}{g_{2,1}e_{2,3}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \alpha_1 := 0_3 \\
\beta_2 &:= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{e_{3,2}} & -\frac{e_{3,3}}{e_{3,2}} \\ 0 & 0 & 1 \end{bmatrix}, \quad \alpha_2 := 0_3 \\
\beta_3 &:= \begin{bmatrix} (z-A_0)B_0^{-1} & -(z-A_0)B_0^{-1}D_0 \\ 0_{1\times 2} & 1 \end{bmatrix} \\
\alpha_3 &:= \begin{bmatrix} -C_0B_0^{-1} & C_0B_0^{-1}D_0 \\ 0_2 & 0_{2\times 1} \end{bmatrix} \\
\beta_4 &:= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{e_{6,3}} \end{bmatrix}, \quad \alpha_4 := 0_3 \\
\beta_5 &:= (z-A_1')B_1'^{-1}, \quad \alpha_5 := -C_1'B_1'^{-1}.
\end{aligned}$$

◇

This can be rewritten as:

$$[\Phi_k\ \Phi_{k+1}] = [\Phi_{k-1}\ \Phi_k]V_k, \quad k = 0, 1, 2, \ldots, 6$$

with

$$V_k := \begin{bmatrix} 0 & \alpha_k \\ I_n & \beta_k \end{bmatrix}.$$

Note that

$$V_k \in \mathbb{R}[z]^{2n\times 2n}, \quad k = 0, 1, 2, \ldots, 5$$

and

$$V_6 \in \mathbb{R}[z]^{2n\times(\underline{n}+1)}.$$

By partitioning these V_k-matrices, one can construct matrix continued fraction formulas for rational forms built up by components of the polynomial vectors ϕ_k.

11 Recurrence relations if all points z_i are on the unit circle

If $|z_i| = 1$, $i = 1, 2, \ldots, m$, then $G := Q^H \Lambda Q$ is a unitary block Hessenberg matrix. This will not influence recurrence relation (7). However, recurrence relation (8) can be rewritten using a decomposition of the matrix G.

Theorem 11.1 (generalized block Schur parameter decomposition) *The unitary block Hessenberg matrix G can be decomposed as*

$$G = G_1 G_2 G_3 \ldots G_{m-\tau_m}$$

with G_i having the form

$$G_i := \begin{bmatrix} I_{k-1} & & \\ & G_i' & \\ & & I_{m-k-1-\lambda_i} \end{bmatrix}$$

with G_i' a unitary $(\lambda_i \times \lambda_i)$-matrix (block Schur parameters) where $\lambda_i := \tau_k + 1$ with k such that $\pi_k' = i$. In the sequel we will also need the following partitioning of G_i'

$$G_j' =: \begin{bmatrix} \gamma_j & \Sigma_j \\ \sigma_j & \Gamma_j \end{bmatrix}, \tag{11}$$

with σ_j a scalar. The entries $\gamma_j, \sigma_j, \Sigma_j, \Gamma_j$ are called the block Schur parameters. The entry $\sigma_{\pi_k'}$ can be read off in the original matrix G, $\sigma_{\pi_k'} = g_{k,\pi_k'}$.
Note that

$$2 \le \lambda_i \le \lambda_j \le n, \quad i < j.$$

Proof.[by induction on i] The unitary block Hessenberg matrix G can be written as:

$$G = G_1 G'.$$

Because the first column of G_1 is equal to the first column of G and because G is unitary, we get that the unitary matrix G' has the form:

$$G' = G_1^H G = \left[\begin{array}{c|c} 1 & 0 \ldots 0 \\ \hline 0 & \\ \vdots & G'' \\ 0 & \end{array}\right].$$

Note that $\sigma_{\pi_k'} = g_{k,\pi_k'}$ with $\pi_k' = 1$. G'' is also unitary and has the same block structure as G, except for the first row and column. Therefore, the same reasoning can be applied again. Note that $g''_{k,\pi_k'} = g_{k,\pi_k'}$, $\pi_k' > 1$. □

Instead of computing the unitary block Hessenberg matrix G, using algorithm 5.1, we construct the blocks G_i', defined by (11), of the block Schur parametrization of G. This reduces the order of computations by a factor m.
Suppose we know the decomposition for m points z_i. Adding one point z_{m+1} with $|z_{m+1}| = 1$, and corresponding weight vector F_{m+1}, gives us the following initial data structure:

$$[\bar{E} \mid \bar{G}] := \left[\begin{array}{c|cc} F_{m+1} & z_{m+1} & 0 \\ E & 0 & G \end{array}\right] \quad \text{with } G = G_1 G_2 \ldots G_{m-\tau_m}.$$

Using unitary similarity transformations, this initial structure is transformed into

$$Q'^H \left[\begin{array}{c|cc} F_{m+1} & z_{m+1} & 0 \\ E & 0 & G \end{array}\right] \begin{bmatrix} I_n & 0 \\ 0 & Q' \end{bmatrix} = [E' \mid G']$$

having zeros under the pivot elements.

Algorithm 11.1 *Transformation of the initial data matrix $\bar{D} := [E \mid G]$ into a matrix having zeros below the pivot elements.*
for $i := 1$ *to* m **do**

- *make element $\bar{d}_{i+1,\pi_i}$ zero by using a Givens rotation (or reflection) J^H with the pivot element (i, π_i):*

$$\bar{E} \leftarrow J^H \bar{E} \tag{12}$$
$$\bar{G} \leftarrow J^H \bar{G} \tag{13}$$

- $\bar{G} \leftarrow \bar{G}J$ *(similarity transformation).*

Note that (12) can be skipped if $r_i = n$.
If $\tau_i < n$ only $n - \tau_i$ nonzero columns of $\bar{E}$ are involved.
Instead of working with the unitary block Hessenberg matrix $\bar{G}$, we work with its decomposition

$$\begin{aligned} \bar{G} &= \begin{bmatrix} z_{m+1} & \\ & I_m \end{bmatrix} \begin{bmatrix} 1 & \\ & G_1 \end{bmatrix} \cdots \begin{bmatrix} 1 & \\ & G_{m-\tau_m} \end{bmatrix} \\ &= \bar{G}_0 \bar{G}_1 \bar{G}_2 \ldots \bar{G}_{m-\tau_m} \end{aligned}$$

which we transform into a decomposition for G'

$$G' = G'_1 G'_2 \ldots G'_{m+1-\tau_{m-1}}$$

Algorithm 11.1 changes as follows.

Algorithm 11.2 *Initilization*

$$\begin{aligned} H_0 &\leftarrow \bar{G}_0 \\ \pi &\leftarrow 0 \end{aligned}$$

for $i := 1$ **to** m **do**
{ *The last pivot element used with $\pi_i > n$ was in column π of $\bar{G}$*}
{$\bar{G} = G'_1 G'_2 \ldots G'_\pi H_{i-1} \bar{G}_i \bar{G}_{i+1} \ldots \bar{G}_{m-\tau_m} \bar{G}_{m-\tau_m+1} \ldots \bar{G}_m$ *with*
$\bar{G}_{m-\tau_m+j} = I_{m+1}, \quad j = 1, 2, \ldots, \tau_m$}
if $1 \le \pi_i \le n$ **then**
* *make element $\bar{e}_{i+1,\pi_i}$ zero by using a Givens rotation (or reflection) J^H with the pivot element $\bar{e}_{i,\pi_i}$:*

$$\begin{aligned} \bar{E} &\leftarrow J^H \bar{E} \\ H_i &\leftarrow J^H H_{i-1} \bar{G}_i J \end{aligned}$$

else *$(\pi_i > n)$*
* *make element $(i+1, \pi_i)$ of H_{i-1} zero by using a Givens rotation (or reflection) J^H with the pivot element (i, π_i) of H_{i-1}:*

$$\begin{aligned} \bar{E} &\leftarrow J^H \bar{E} \\ G'_{\pi+1} H_i &\leftarrow J^H H_{i-1} \bar{G}_i J, \quad \pi \leftarrow \pi + 1 \end{aligned}$$

{ *I.e. $G'_{\pi+1}$ is the first block Schur parameter of $J^H H_{i-1} \bar{G}_i J$, while H_i is the tail of the generalized block Schur decomposition* }
$G'_{m+1-\tau_{m+1}} \leftarrow H_m$

Note that in the else-part, the elements $(i+1, \pi_i)$ and (i, π_i) of H_{i-1} are also the elements at the same position in $\bar{G}$.
For notational simplicity, we have written down the algorithm using $(m+1) \times (m+1)$ matrices. However, when looking at the computational complexity, we have only to take into consideration the nontrivial operations. Besides constructing the m Givens rotations, we have the step $\bar{E} \leftarrow J^H \bar{E}$ involving $\leq 4n$ multiplications. The nontrivial part of $J^H H_{i-1} \bar{G}_i J$ is a unitary matrix of size $\leq (2n+2) \times (2n+2)$. Therefore, adding one new data point (z_{m+1}, F_{m+1}) requires a number of multiplications proportional to m and not to m^2 as in the general case. Therefore, constructing $[E \mid G]$ for m data points needs a number of multiplications proportional to m^2. Hence, the amount of computational work, like in the real case, is also reduced by an order of magnitude m. Note that if we are only interested in $\phi_{|\Delta|}$, the computational work is proportional to $m|\Delta|$.

Once we have computed E and $G_1, G_2, \ldots, G_{m-\tau_m}$, we have the following recurrence relations for the columns Q_k of Q, $k = 1, 2, 3, \ldots, m$:

a) $1 \leq \pi_k \leq n$:

$$e_{k,\pi_k} Q_k = F'_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k} Q_i \quad \text{(see (5))} \tag{14}$$

b) $\pi_k - n =: \pi'_k > 0$:
We know that: $\Lambda Q = Q G_1 G_2 \ldots G_{m-\tau_m}$.

For $k = 1, 2, \ldots, m$, we define $Q_1^{(k)}, Q_2^{(k)}, \ldots, Q_k^{(k)}$ as

$$[Q_1^{(k)}\ Q_2^{(k)} \ldots Q_k^{(k)}\ Q_{k+1}^{(k)}\ Q_{k+2}^{(k)} \ldots Q_m^{(k)}] := Q G_1 G_2 \ldots G_{\pi'_j}$$

with

$$\pi'_j := \max\{\pi'_i | i \leq k\}.$$

Note that if $\pi_k - n =: \pi'_k > 0$, we have

$$[Q_1^{(k-1)}\ Q_2^{(k-1)} \ldots Q_m^{(k-1)}] = [Q_1^{(j)}\ Q_2^{(j)} \ldots Q_j^{(j)}\ Q_{j+1} \ldots Q_k\ Q_{k+1} \ldots Q_m].$$

Multiplying the previous columns by $G_{\pi'_k}$, we get

$$\begin{aligned} Q G_1 G_2 \ldots G_{\pi'_k - 1} G_{\pi'_k} &= [Q_1^{(k-1)} \ldots Q_{k-1}^{(k-1)}\ Q_k \ldots Q_m] G_{\pi'_k} \\ &= [Q_1^{(k)} \ldots Q_{k-1}^{(k)}\ Q_k^{(k)}\ Q_{k+1} \ldots Q_m]. \end{aligned} \tag{15}$$

If we partition the nontrivial $(\tau_j + 1) \times (\tau_j + 1)$ part G'_j of G_j (see (11)) using the block Schur parameters as

$$G'_j =: \begin{bmatrix} \gamma_j & \Sigma_j \\ \sigma_j & \Gamma_j \end{bmatrix},$$

with σ_j 1×1, we can rewrite (15) as:

$$[Q_{\pi'_k}^{(k-1)}\ Q_{\pi'_k+1}^{(k-1)} \ldots Q_{k-1}^{(k-1)}\ Q_k] \begin{bmatrix} \gamma_{\pi'_k} & \Sigma_{\pi'_k} \\ \sigma_{\pi'_k} & \Gamma_{\pi'_k} \end{bmatrix} = [Q_{\pi'_k}^{(k)}\ Q_{\pi'_k+1}^{(k)} \ldots Q_{k-1}^{(k)}\ Q_k^{(k)}]. \tag{16}$$

Remember from theorem 11.1 that $\sigma_{\pi'_k} = g_{k,\pi'_k}$. Because the π'_k-th column of

$$\Lambda Q = Q G_1 G_2 \ldots G_{m-\tau_m}$$

is equal to the π'_k-th column of $Q G_1 G_2 \ldots G_{\pi'_k}$, we get the following recurrence relation for Q_k by taking the first column of the left hand side of (16):

$$\sigma_{\pi'_k} Q_k = \Lambda Q_{\pi'_k} - [Q_{\pi'_k}^{(k-1)}\ Q_{\pi'_k+1}^{(k-1)} \ldots Q_{k-1}^{(k-1)}] \gamma_{\pi'_k}. \tag{17}$$

In the next steps, we do not need $Q^{(k)}_{\pi'_k}$ anymore. Therefore, by taking the last columns of left and right hand side of (16), we get the following recurrence relation for the auxiliary columns $Q^{(k)}_j$, $j = \pi'_k + 1, \ldots, k$:

$$[Q^{(k)}_{\pi'_k+1} \cdots Q^{(k)}_k] = [Q^{(k-1)}_{\pi'_k} \cdots Q^{(k-1)}_{k-1}\ Q_k] \left[\begin{array}{c} \Sigma_{\pi'_k} \\ \Gamma_{\pi'_k} \end{array} \right]. \tag{18}$$

The recurrence relations (14), (17) and (18) can be rewritten as recurrence relations with a limited number of terms. We are immediately rewriting this in terms of the orthonormal polynomial vectors ϕ_i. For each $k = 1, 2, \ldots, m$, we start with

$$[\phi_{k-n}\ \phi_{k-n+1} \cdots Q_{k-1} || \underbrace{\phi^{(k-1)}_{k-\tau_k} \cdots \phi^{(k-1)}_{k-1}}_{\tau_k} | S^{(k-1)}_{\tau_k+1} \cdots S^{(k-1)}_n]$$

with

$$S^{(k-1)}_j := U_j - \sum_{i=1}^{k-1} e_{i,j} \phi_i, \quad j = \tau_k + 1, \tau_k + 2, \ldots, n.$$

If $1 \leq \pi_k \leq n$, we can use recurrence relation (14) to get

$$\begin{array}{rl} & [\phi_{k-n+1} \cdots \phi_{k-1} | \phi_k || \phi^{(k)}_{k-\tau_k} \cdots \phi^{(k)}_{k-1} | \phi^{(k)}_k | S^{(k)}_{\tau_k+2} \cdots S^{(k)}_n] \\ \leftarrow & [\phi_{k-n+1} \cdots \phi_{k-1} || \phi^{(k-1)}_{k-\tau_k} \cdots \phi^{(k-1)}_{k-1} | S^{(k-1)}_{\tau_k+1} | S^{(k-1)}_{\tau_k+2} \cdots S^{(k-1)}_n]\ T_k \end{array}$$

with

$$T_k := \left[\begin{array}{c|c||c|c|c} D_n & 0 & 0 & 0 & 0 \\ \hline\hline 0 & 0 & I_{\tau_k} & 0 & 0 \\ \hline 0 & \frac{1}{e_{k,\pi_k}} & 0 & \frac{1}{e_{k,\pi_k}} & -\frac{1}{e_{k,\pi_k}}[e_{k,\pi_k+1} \cdots e_{k,n}] \\ \hline 0 & 0 & 0 & 0 & I_{n-\tau_k-1} \end{array} \right]$$

with

$$D_j := \left[\begin{array}{cccc} 0 & 0 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ & & & 0 \\ 0 & 0 & \ldots & 1 \end{array} \right] \in \mathbb{C}^{j \times (j-1)}.$$

If $\pi_k - n =: \pi'_k > 0$, we can use recurrence relations (17) and (18) to get:

$$\begin{array}{rl} & [\phi_{k-n+1} \cdots \phi_{k-1} | \phi_k || \phi^{(k)}_{k-\tau_k+1}\ \phi^{(k)}_{k-\tau_k+2} \cdots \phi^{(k)}_k | S^{(k)}_{\tau_k+1} \cdots S^{(k)}_n] \\ \leftarrow & [\phi_{k-n} \cdots | \phi_{\pi'_k} | \cdots \phi_{k-1} || \phi^{(k-1)}_{k-\tau_k} \cdots \phi^{(k-1)}_{k-1} | S^{(k-1)}_{\tau_k+1} \cdots S^{(k)}_n]\ T_k \end{array}$$

with

$$
T_k := \left[\begin{array}{c|c||c|c}
 & 0 & 0 & 0 \\ \hline
D_n & \frac{z}{\sigma_{\pi'_k}} & \frac{z}{\sigma_{\pi'_k}}\Gamma_{\pi'_k} & \frac{-z}{\sigma_{\pi'_k}}[e_{k,\tau_k+1}\dots e_{k,n}] \\ \hline
 & 0 & 0 & 0 \\ \hline\hline
0 & \frac{-\gamma_{\pi'_k}}{\sigma_{\pi'_k}} & \Sigma_{\pi'_k} - \frac{\gamma_{\pi'_k}}{\sigma_{\pi'_k}}\Gamma_{\pi'_k} & \frac{\gamma_{\pi'_k}}{\sigma_{\pi'_k}}[e_{k,\tau_k+1}\dots e_{k,n}] \\ \hline
0 & 0 & 0 & I_{n-\tau_k}
\end{array}\right].
$$

Note that

$$\Sigma_{\pi'_k} - \gamma_{\pi'_k}\sigma_{\pi'_k}^{-1}\Gamma_{\pi'_k} = \Sigma_{\pi'_k}^{-H}$$

and

$$\sigma_{\pi'_k}^{-1}\Gamma_{\pi'_k} = -\gamma_{\pi'_k}^{H}\Sigma_{\pi'_k}^{-H}.$$

Hence, looking at the second and third block column of T_k, we see a type of Szegő recurrence relations.

These recurrence relations can be combined to get generalized block Szegő recurrence relations. To avoid notational complexity, we only give an example of this result.

Example 11.1 Suppose the transformed data matrix has the following structure:

$$
[E \mid G] = \left[\begin{array}{ccc|cccccccc}
\circledast & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \circledast & \times & \times & \times & \times & \times & \times & \times \\
 & \circledast & \times & & \times & \times & \times & \times & \times & \times & \times \\
 & & \times & & \circledast & \times & \times & \times & \times & \times & \times \\
 & & \circledast & & & \times & \times & \times & \times & \times & \times \\
 & & & & & \circledast & \times & \times & \times & \times & \times \\
 & & & & & & \circledast & \times & \times & \times & \times \\
 & & & & & & & \circledast & \times & \times & \times
\end{array}\right].
$$

Let us define

$$
\begin{array}{ll}
\Phi_0 := I_3 & \Phi'_0 := 0_3 \\
\Phi_1 := [\phi_1\ S_2^{(1)}\ S_3^{(1)}] & \Phi'_1 := [\phi_1^{(1)}\ 0\ 0] \\
\Phi_2 := [\phi_2\ S_2^{(2)}\ S_3^{(2)}] & \Phi'_2 := [\phi_2^{(2)}\ 0\ 0] \\
\Phi_3 := [\phi_2\ \phi_3\ S_3^{(3)}] & \Phi'_3 := [\phi_2^{(3)}\ \phi_3^{(3)}\ 0] \\
\Phi_4 := [\phi_3\ \phi_4\ \phi_5] & \Phi'_4 := [\phi_3^{(5)}\ \phi_4^{(5)}\ \phi_5^{(5)}] \\
\Phi_5 := [\phi_6\ \phi_7\ \phi_8] & \Phi'_5 := [\phi_6^{(8)}\ \phi_7^{(8)}\ \phi_8^{(8)}]
\end{array}
$$

The polynomial matrices Φ_k satisfy the generalized block Szegő recurrence relation

$$[\Phi_k\ \Phi'_k] = [\Phi_{k-1}\ \Phi'_{k-1}]\begin{bmatrix} A_{k-1} & C_{k-1} \\ B_{k-1} & D_{k-1}\end{bmatrix} \quad , \quad k = 0, 1, 2, 3, 4,$$

with

$$A_0 := \begin{bmatrix} \frac{1}{e_{1,1}} & \frac{-e_{1,2}}{e_{1,1}} & \frac{-e_{1,3}}{e_{1,1}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad B_0 := 0_3$$

$$C_0 := 0_3 \qquad D_0 := 0_3$$

$$A_1 := \begin{bmatrix} \frac{z}{\sigma_1} & \frac{-z}{\sigma_1}e_{2,2} & \frac{-z}{\sigma_1}e_{2,3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B_1 := \begin{bmatrix} \frac{-\gamma_1}{\sigma_1} & \frac{-\gamma_1}{\sigma_1}e_{2,2} & \frac{-\gamma_1}{\sigma_1}e_{2,3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C_1 := \begin{bmatrix} \frac{z}{\sigma_1}\Gamma_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad D_1 := \begin{bmatrix} \Sigma_1^{-H} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{e_{3,2}} & \frac{-e_{3,3}}{e_{3,2}} \\ 0 & 0 & 1 \end{bmatrix} \qquad B_2 := 0_3$$

$$C_2 := \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{e_{3,2}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad D_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_3 := \begin{bmatrix} 0 & \frac{z}{\sigma_2} & \frac{-ze_{4,3}}{\sigma_2 e_{5,3}} \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad B_3 := \begin{bmatrix} 0 & \frac{-\gamma_2}{\sigma_2} & \frac{\gamma_2}{\sigma_2}\frac{e_{4,3}}{e_{5,3}} \\ 0 & 0 & 0 \end{bmatrix}$$

$$C_3 := \begin{bmatrix} \frac{z}{\sigma_2}\Gamma_2 & \frac{-ze_{4,3}}{\sigma_2 e_{5,3}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad D_3 := \begin{bmatrix} \Sigma_2^{-H} & \frac{\gamma_2 e_{4,3}}{\sigma_2 e_{5,3}} \\ 0 & 0 \end{bmatrix}$$

$$A_4 := z\ \sigma_{3,5}^{-1} \qquad B_4 := -\gamma_{3,5}\sigma_{3,5}^{-1}$$

$$C_4 := -\gamma_{3,5}\Sigma_{3,5}^{-H} \qquad D_4 := \Sigma_{3,5}^{-H}$$

with

$$\begin{bmatrix} G_3 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & G_4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & G_5 \end{bmatrix} =: \begin{bmatrix} \gamma_{3,5} & \Sigma_{3,5} \\ \sigma_{3,5} & \Gamma_{3,5} \end{bmatrix}.$$

Note that the last block recurrence relation is just the classical block Szegő recurrence relation. If we add more data points, we can use the latter relation to compute the next block of 3 orthonormal polynomial vectors ϕ_i. ◇

12 Conclusion

In this paper, we have constructed several variants of an algorithm which computes the coefficients of recurrence relations for orthonormal polynomial vectors with respect to a discrete inner product. When the points z_i are real or on the unit circle, we have shown that the number of computations reduces an order of magnitude. Also the recurrence relations only require a fixed number of terms.

The orthonormal polynomial vectors were used to solve a discrete least squares approximation problem. Future work will show further applications of these orthonormal polyunomial vectors.

References

[1] G.S. Ammar and W.B. Gragg. $O(n^2)$ reduction algorithms for the construction of a band matrix from spectral data. *SIAM J. Matrix Anal. Appl.*, 12:426–431, 1991.

[2] G.S. Ammar, W.B. Gragg, and L. Reichel. Constructing a unitary Hessenberg matrix from spectral data. In G. Golub and P. Van Dooren, editors, *Numerical linear algebra,*

digital signal processing and parallel algorithms, volume 70 of *NATO-ASI Series, F: Computer and Systems Sciences*, pages 385–395, Berlin, 1991. Springer-Verlag.

[3] D.L. Boley and G.H. Golub. A survey of matrix inverse eigenvalue problems. *Inverse Problems*, 3:595–622, 1987.

[4] A. Bultheel and M. Van Barel. Vector orthogonal polynomials and least squares approximation. *SIAM Review*, 1993. Submitted. Also report TW184, Department of Computing Science, K.U.Leuven.

[5] P. Delsarte, Y. Genin, and Y. Kamp. Orthogonal polynomial matrices on the unit circle. *IEEE Trans. Circuits and Systems*, 25:149–160, 1978.

[6] V.K. Dubovoj, B. Fritzsche, and B. Kirstein. *Matricial version of the classical Schur problem*, volume 129 of *Teubner-Texte zur Mathematik*. Teubner Verlagsgesellschaft, Stuttgart, Leipzig, 1992.

[7] S. Elhay, G.H. Golub, and J. Kautsky. Updating and downdating of orthogonal polynomials with data fitting applications. *SIAM J. Matrix Anal. Appl.*, 12:327–353, 1991.

[8] J.S. Geronimo. Matrix orthogonal polynomials on the unit circle. *J. Math. Phys.*, 22(7):1359–1365, 1981.

[9] J.S. Geronimo. Scattering theory and matrix orthogonal polynomials on the real line. *Circuits, Systems and Signal Process.*, 1:471–495, 1982.

[10] W. B. Gragg and W. J. Harrod. The numerically stable reconstruction of Jacobi matrices from spectral data. *Numer. Math.*, 44:317–335, 1984.

[11] L. Reichel. Fast QR decomposition of Vandermonde-like matrices and polynomial least squares approximation. *SIAM J. Matrix Anal. Appl.*, 12:552–564, 1991.

[12] L. Reichel, G.S. Ammar, and W.B. Gragg. Discrete least squares approximation by trigonometric polynomials. *Math. Comp.*, 57:273–289, 1991.

[13] H. Rutishauser. On Jacobi rotation patterns. In *Proceedings of Symposia in Applied Mathematics, vol. 15, Experimental Arithmetic, High Speed Computing and Mathematics*, pages 219–239, Providence, 1963. Amer. Math. Society.

[14] M. Van Barel and A. Bultheel. A parallel algorithm for discrete least squares rational approximation. *Numer. Math.*, 63:99–121, 1992.

[15] M. Van Barel and A. Bultheel. Discrete least squares rational approximation on the unit circle. *J. Comput. Appl. Math.*, 1993. To appear, November 1993. Also report TW135, Department of Computing Science, K.U.Leuven.