Electronic Transactions on Numerical Analysis.
Volume 3, pp. 1-23, March 1995.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

ORTHONORMAL POLYNOMIAL VECTORS AND LEAST SQUARES APPROXIMATION FOR A DISCRETE INNER PRODUCT*

M. VAN BAREL AND A. BULTHEEL †

Abstract. We give the solution of a discrete least squares approximation problem in terms of orthonormal polynomial vectors with respect to a discrete inner product. 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 weight vectors 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. As an application, we approximate a vector-valued function by a vector rational function in a linearized least squares sense.

Key words. orthonormal polynomial vectors, least squares approximation, vector rational approximation.

AMS subject classifications. 42C05, 30E10, 65D10, 41A28, 41A20.

1. Introduction.

Suppose we want to approximate a set of data points $(z_i, f_i, e_i) \in \mathbb{C}^3$, $i = 1, 2, \ldots, m$, by a rational function $n(z)/d(z)$ of a given degree structure in the sense that we want to minimize the sum of squares S with

$$S := \sum_{i=1}^{m} |f_i/e_i - n(z_i)/d(z_i)|^2.$$

This is a highly nonlinear problem which requires an iterative solver. A good starting value for the iteration can often be obtained by solving the linearized problem, where we minimize S with

$$S := \sum_{i=1}^{m} |f_i d(z_i) - e_i n(z_i)|^2.$$

The latter problem is linear and much easier to solve. If we set

$$F_i^H := [f_i, -e_i] \qquad \text{and } P(z)^T := [d(z), n(z)],$$

we can rewrite the latter as

$$S = \sum_{i=1}^{m} |F_i^H P(z_i)|^2.$$

We can use the same setup if, at each knot z_i, a vector of complex data is given, i.e., if $f_i \in \mathbb{C}^{n-1}$, $e_i \in \mathbb{C}$, this vector is approximated by a vector rational function, so that also $n(z)$ is an $(n-1)$-dimensional vector polynomial and $d(z)$ a common (scalar) denominator. In the last form of S, we then have $F_i \in \mathbb{C}^{n\times 1}$ and $P(z) \in \mathbb{C}[z]^{n\times 1}$. Of course we have degree conditions on $P(z)$, say

$$\partial P \le \Delta := [\delta_1, \delta_2, \ldots, \delta_n] \text{ (componentwise)}, \qquad \Delta \in (\mathbb{N} \cup \{-1\})^{n\times 1}$$

* Received for publication September 26, 1994. Accepted for publication January 27, 1995. Communicated by G. Ammar.

† Department of Computer Science, K.U. Leuven, Celestijnenlaan 200 A, 3001 Heverlee, Belgium

ETNA
Kent State University
etna@mcs.kent.edu

(we say that the zero polynomial has degree -1). To avoid the trivial solution $P \equiv 0$, 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

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

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 by an order of magnitude.

In previous publications [5, 16, 17], we have considered special cases of the approximation problem described above. In [16], 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 [12], which constructs the optimal polynomial fitting given function values in real points z_i in a least squares sense. Reichel's algorithm itself is based on the Rutishauser-Gragg-Harrod algorithm [14, 11, 1] for the computation of Jacobi matrices. Similar results were obtained in [4, 8]. In Section 9, 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 [13] solve the approximation problem when the given function values are taken in points on the unit circle. In [17], we generalized this from $n = 1$ to $n = 2$ with equal degrees $\delta_1 = \delta_2$. Section 10 handles the general problem on the unit circle. When $n = 2$, we refer the reader to [5], which summarizes [16] and [17] and handles the case of arbitrary degrees δ_1 and δ_2.

In [16, 17], we have given 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 7, we give the conditions for having an interpolating polynomial vector. In a future 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. For the simpler problem of vector rational approximation, we give an example in Section 11.

2. Discrete least squares approximation problem. We consider the following inner product.

DEFINITION 2.1 (INNER PRODUCT, NORM). *Given the points $z_i \in \mathbb{C}$, and the weight vectors $F_i \in \mathbb{C}^{n\times 1}$, $i = 1, 2, \ldots, m$, we consider a subspace $\mathcal{P}$ of all polynomial vectors $\mathbb{C}[z]^{n\times 1}$ such that the following bilinear form defines a discrete inner product $\langle P, Q \rangle$ for two polynomial vectors $P, Q \in \mathcal{P} \subset \mathbb{C}[z]^{n\times 1}$:*

$$\langle P, Q \rangle := \sum_{i=1}^{m} P^H(z_i) F_i F_i^H Q(z_i). \tag{2.1}$$

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

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

For this to be an inner product in $\mathcal{P}$, it is necessary and sufficient that $\mathcal{P}$ is a subspace of polynomial vectors such that there is no nonzero polynomial vector $P \in \mathcal{P}$ with

ETNA
Kent State University
etna@mcs.kent.edu

$\langle P, P\rangle = 0$ or equivalently with $F_i^H P(z_i) = 0$, $i = 1, 2, \ldots, m$. We will call this the *regular case*. In Section 7, we will handle the *singular* case. Up to then, we will assume that (2.1) is a (positive definite) inner product. 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 $F_i \in \mathbb{C}^{n\times 1}$, $i = 1, 2, \ldots, m$, 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 $\mathcal{P}_{\bar{\Delta}}$*

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

In the discrete least squares approximation problem, we look for the polynomial vector P such that $\|P\| = \min_{Q\in\mathcal{P}_{\bar{\Delta}}} \|Q\|$. The degree vector Δ is such that for the set $\mathcal{P}_\Delta$ $\|\cdot\|$ is a norm (not a semi-norm), i.e. $\mathcal{P}_\Delta \subset \mathcal{P}$.

Note that in this paper, all inequalities between integer vectors are taken componentwise.

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

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

in a least squares sense, i.e.

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

(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 $B_j := [B_{j,1}, B_{j,2}, \ldots, B_{j,n}]^T$ 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 chosen freely. We get

$$\begin{aligned} \|P\|^2 &= \langle P, P\rangle \\ &= \left\langle \sum_{j=1}^{|\Delta|} B_j a_j, \sum_{j=1}^{|\Delta|} B_j a_j \right\rangle \end{aligned}$$

ETNA
Kent State University
etna@mcs.kent.edu

$$= \sum_{j=1}^{|\Delta|} |a_j|^2 \quad \text{(because } \langle B_i, B_j \rangle = \delta_{ij}).$$

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).

Suppose we want to solve the problem for a certain degree vector $\Delta^\star$. We want to construct the basis vectors for $\mathcal{P}_{\Delta^\star}$ recursively by gradually constructing the basis vectors for nested subspaces

$$\mathcal{P}_{\Delta^{(0)}} \subset \mathcal{P}_{\Delta^{(1)}} \subset \cdots \subset \mathcal{P}_{\Delta^{(k)}} = \mathcal{P}_{\Delta^\star}.$$

If we arrange the degree vectors $\Delta^{(k)}$ into an n-dimensional table, then we want to reach $\Delta^\star$ by walking along a "diagonal". This means that we pass through the points $\Delta^\star - U, \Delta^\star - 2U, \ldots$, where $U := [1, 1, \ldots, 1]^T$. Each move on the diagonal from Δ to $\Delta + U$ will be decomposed in a set of n elementary steps in each of the coordinate directions: $\Delta + U_1^1, \Delta + U_2^1, \ldots, \Delta + U_n^1 = \Delta + U$, where $U_j^1 := [1, 1, \ldots, 1, 0, \ldots, 0]^T$ (j ones). This results in a staircase-like polyline. This works quite well when $\Delta \geq 0$. Unless $\Delta^\star$ is on the main diagonal, the starting point of the diagonal through $\Delta^\star$ will be outside the positive part of the coordinate system. When 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 $k \geq 0$. This means that $\dim \mathcal{P}_{\Delta^{(k+1)}} = \dim \mathcal{P}_{\Delta^{(k)}} + 1$, starting with $\Delta^{(0)} = [-1, -1, \ldots, -1]^T$, which corresponds to $\mathcal{P}_{\Delta^{(0)}} = \{[0, 0, \ldots, 0]^T\}$ with $\dim \mathcal{P}_{\Delta^{(0)}} = 0$. Hence, we have the following definition for the sequence of degree vectors $\Delta^{(k)}$, degree indices ν_k and the so-called pivot indices π_k, $k = 1, 2, \ldots$.

Definition 3.1 (degree vectors, degree indices and pivot indices). *Let $\Delta^\star := [\delta_1^\star, \ldots, \delta_n^\star]$ be the target degree vector and define $U_j := [0, \ldots, 0, 1, 0, \ldots, 0]^T$ (1 at the j-th position). Set the initial degree vector $\Delta^{(0)} := [-1, -1, \ldots, -1]^T$. Furthermore, if $\Delta^{(k-1)} = [\delta_1, \delta_2, \ldots, \delta_n]^T$, then $\Delta^{(k)} := \Delta^{(k-1)} + U_j$ with j the least integer in $\{1, 2, \ldots, n\}$ that satisfies the equation $\delta_j^\star - \delta_j = \max\{\delta_i^\star - \delta_i | i = 1, 2, \ldots, n\}$. The corresponding degree index is $\nu_k := j$. The pivot indices π_k are defined as follows. If $\delta_j = -1$ then $\pi_k := j$. Otherwise $\pi_k := k - l + n$ with l the number of nonnegative elements in $\Delta^{(k)}$.*

By defining the degree vectors in this way, for each degree vector $\Delta^{(k)} > -U$ the degree vector $\Delta^{(k)} - U$ appears earlier in the sequence as $\Delta^{(j)} = \Delta^{(k)} - U$. Also $k - j = n$ is made as small as possible. This will result in a recurrence relation for the orthonormal polynomial vector $\phi_k(z)$ written as a linear combination of $z\phi_j(z)$ and the other previous orthonormal polynomial vectors. However, when all points z_i are on the real line or on the unit circle, only a limited number of the previous orthonormal polynomial vectors will be needed. Hence, the computational work will be decreased by an order of magnitude.

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 with the prescribed degree structure.

4. Algorithm. In this section, we give an algorithm which inputs the initial data (the points z_i, the weights F_i) and outputs the building blocks of a recurrence relation generating the desired orthonormal polynomial vectors. This transformation process is influenced by the parameters

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

with

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

Note that by a permutation this ordering can always be assumed without loss of generality.

The algorithm starts with the following matrix:

$$\left[\begin{array}{c|cccc} F_1 & z_1 & & & \\ F_2 & & z_2 & & \\ \vdots & & & \ddots & \\ F_m & & & & z_m \end{array}\right] \quad =: \quad \left[\begin{array}{c|c} F & \Lambda \end{array}\right] \in \mathbb{C}^{m\times(n+m)},$$

and transforms this using similarity transformations on Λ into

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

(Q unitary) such that $\left[Q^H F \mid Q^H \Lambda Q\right]$ has zeros below the pivot positions (i, π_i), $i = 1, 2, \ldots, m$. The following algorithm will add, for each i, the point z_i with corresponding weight F_i. Note also, that each iteration changes the underlying inner product.

Algorithm 4.1. *Transformation of the initial data matrix* $D := [F \mid \Lambda] =: [d_{ij}]$ *into a matrix* $\left[Q^H F \mid Q^H \Lambda Q\right]$ *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 4.1 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 vectors is therefore bounded 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 4.1 could be designed.

5. Recurrence relations for the columns of the unitary transformation matrix Q. In the previous section we 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

$$F = QE, \tag{5.1}$$

$$\Lambda Q = QG. \tag{5.2}$$

Knowing $E =: [e_{i,j}]$ and $G =: [g_{i,j}]$, we can reconstruct the columns Q_k of Q, $k = 1, 2, 3, \dots, m$ 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 the pivot position (k, π_k). Therefore, writing out equality (5.1) for the π_k-th column gives us (F'_j denotes the j-th column of F)

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

with

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

So, we can write Q_k as

$$e_{k,\pi_k} Q_k = F'_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k} Q_i. \tag{5.3}$$

b) $\pi_k - n =: \pi'_k > 0$. We know that $g_{i,\pi'_k} = 0$, $i > k$. Writing out equality (5.2) 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

$$g_{k,\pi'_k} Q_k = \Lambda Q_{\pi'_k} - \sum_{i=1}^{k-1} g_{i,\pi'_k} Q_i. \tag{5.4}$$

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

$$\tau_k := k - \pi'_k = \#\{\pi_j | 1 \leq \pi_j \leq n, j < k\}. \tag{5.5}$$

As long as e_{k,π_k} and g_{k,π'_k} are different from zero, we can use (5.3) and (5.4) as a recurrence relation to compute the columns $Q_k,\ k = 1, 2, 3, 4, \ldots$. In Section 7, we shall see that e_{k,π_k} and g_{k,π'_k} will be nonzero in the regular case.

6. Recurrence relations for a sequence of orthonormal polynomial vectors. Similar to the recurrence relations (5.3) and (5.4) for the columns Q_k, we can construct a sequence of polynomial vectors $\{\phi_k\}_{k=1}^m$, $\phi_k \in \mathbb{C}[z]^{n\times 1}$ as follows:

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

$$e_{k,\pi_k}\phi_k(z) = U_{\pi_k} - \sum_{i=1}^{k-1} e_{i,\pi_k}\phi_i(z) \tag{6.1}$$

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

$$g_{k,\pi'_k}\phi_k(z) = z\phi_{\pi'_k}(z) - \sum_{i=1}^{k-1} g_{i,\pi'_k}\phi_i(z). \tag{6.2}$$

THEOREM 6.1 (RELATIONSHIP BETWEEN Q_k AND $\phi_k(z)$). *Let F_k denote the rows of F and F'_k the columns of F:*

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

Then

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

with

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

Proof. The result it true for $k = 1$, because $(U_1 = [1, 0, \ldots, 0]^T)$

$$\begin{aligned} e_{1,1}Q_1 &= F'_1 = F^\star \begin{bmatrix} U_1 \\ U_1 \\ \vdots \\ U_1 \end{bmatrix}, \\ e_{1,1}\phi_1(z) &= U_1, \quad \text{hence} \quad e_{1,1}\phi_1^\star = \begin{bmatrix} U_1 \\ U_1 \\ \vdots \\ U_1 \end{bmatrix}. \end{aligned}$$

Thus,

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

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

ETNA
Kent State University
etna@mcs.kent.edu

a) $1 \leq \pi_k \leq n$: Take the recurrence relation (5.3) 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 6.2 (ORTHONORMALITY OF ϕ_k). *The polynomial vectors, defined by (6.1) and (6.2), satisfy*

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

where the inner product is defined in (2.1).

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

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

□

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 pivot indices $\pi_1, \pi_2, \ldots$ indeed gives the desired degree structure $(\Delta^{(k)}, \nu_k)$ of the orthonormal polynomial vectors ϕ_k.

THEOREM 6.3. *The orthonormal polynomial vectors ϕ_k computed by Algorithm 4.1 have the corresponding extended degree vectors $\bar{\Delta}^{(k)} = (\Delta^{(k)}, \nu_k)$; i.e.,*

- $\partial \phi_k \leq \Delta^{(k)}$;
- *the ν_k-th component of ϕ_k is non-zero.*

Proof. We proceed by induction on k. It is clear that the theorem is true for $k = 1$. Suppose the theorem is true for $1, 2, \ldots, k-1$. When the recurrence relation (6.1) is used, $\phi_k(z)$ has the extended degree vector $\bar{\Delta}^{(k)}$ because $\Delta^{(i)} \leq \Delta^{(k)}$ and the π_k-th component is equal to -1, $i = 1, 2, \ldots, k-1$.

When the recurrence relation (6.2) is used, we see from (5.5) that only the first τ_k components of $\Delta^{(i)}, i = 1, 2, \ldots, k$, are greater than -1. Hence, $\Delta^{(\pi'_k)} = \Delta^{(k)} - U^1_{\tau_k}$ and $\nu_{\pi'_k} = \nu_k$. The degree vectors $\Delta^{(i)}, i = 1, 2, \ldots, k-1$, are smaller than or equal to $\Delta^{(k)} - U_{\nu_k}$. Therefore, using recurrence relation (6.2) gives an orthonormal polynomial vector having the desired degree structure. □

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 (6.1) and (6.2). Therefore, Algorithm 4.1 can be adapted to compute only 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.

7. Singular case. Until now, we assumed all the entries e_{k,π_k} and g_{k,π'_k}, $k = 1, 2, \ldots m$ to be different from zero. In this case, all orthonormal polynomial vectors $\phi_k(z)$ can be computed by using the recurrence relations (6.1) and (6.2). For each k, $1 \leq k \leq m$, the inner product is a true inner product (positive definite). Hence, the subspace $\mathcal{P}_{\Delta^{(k)}} \subset \mathcal{P}$; i.e., we are in the regular case. Indeed, each polynomial vector $P \in \mathcal{P}_{\Delta^{(k)}}$ can be written as a linear combination of the orthonormal polynomial vectors:

$$P = \sum_{i=1}^{k} a_i \phi_i.$$

Hence, $\|P\|^2 = \sum_{i=1}^{k} |a_i|^2$. This can only be zero when $P \equiv 0$.

Suppose now that some of the entries e_{k,π_k} or g_{k,π'_k}, $k = 1, 2, \ldots m$ are zero. Suppose that the first entry equal to zero is

1. e_{k,π_k} : In this case, we cannot use recurrence relation (6.1) 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 (5.3), 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, $F_j \phi'_k(z_j) = 0$, $j = 1, 2, \ldots, m$.

2. g_{k,π'_k}: As in 1., 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.

ETNA
Kent State University
etna@mcs.kent.edu

8. 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 with

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

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:

$$\langle P,Q\rangle := \sum_{k=1}^{m} P(z_k)^H F_k^H F_k Q(z_k) \in \mathbb{C}^{l\times l}, \tag{8.1}$$

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

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

using the orthonormal polynomial vectors $\phi_k(z)$ where $U_j^0 := [0,0,\ldots,0,1,\ldots,1]^T$ (j zeros). By grouping together l of these orthonormal polynomial vectors, we get (a kind of) orthonormal polynomial matrices with respect to (8.1).

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 by 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 for example [6, 9, 10, 7].

9. 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 $g_{k,j} = 0$ for $j < \pi_k$, we also have $g_{j,k} = 0$ for $j < \pi_k$. The recurrence relation (6.1) to compute a sequence of orthonormal polynomial vectors will not change in this case, but recurrence relation (6.2) will have a smaller number of terms in the right-hand side:

$$g_{k,\pi'_k} \phi_k(z) = z\phi_{\pi'_k(z)} - \sum_{i=\lambda_k}^{k-1} g_{i,\pi'_k} \phi_i(z). \tag{9.1}$$

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 (9.1) 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.

1. When $n = 1$ (the scalar case), the recurrence relation (9.1) 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

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

2. When $\pi_i = i$, $i = 1, 2, \ldots, n$, we use recurrence relation (6.1) to compute $\phi_1, \phi_2, \ldots, \phi_n$. For $k > n$, recurrence relation (9.1) 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 4.1 reduces by an order of magnitude in case all z_i are real. Each Givens rotation (or reflection) involves vectors of length at most $2(n+1)$ instead of vectors of length $i+n+1-j$. Applying the Givens rotation to the left requires at most $8(n+1)$ multiplications. Applying the Givens rotation to the right requires only 8 multiplications because of symmetry considerations. Therefore, the total number of multiplications is bounded by

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

ETNA
Kent State University
etna@mcs.kent.edu

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. Due to the notational complexity, we only give an example indicating this equivalence.

EXAMPLE 9.1. Suppose the transformed data matrix has the following structure:

$$
\begin{aligned}
[E \;\; G] \;&=\; \left[\begin{array}{ccc|ccccccc}
\circledast & \times & \times & \times & \times & 0 & 0 & 0 & 0 & 0 \\
0 & \times & \times & \circledast & \times & \times & \times & 0 & 0 & 0 \\
0 & \circledast & \times & 0 & \times & \times & \times & \times & 0 & 0 \\
0 & 0 & \times & 0 & \circledast & \times & \times & \times & \times & \times \\
0 & 0 & \times & 0 & 0 & \circledast & \times & \times & \times & \times \\
0 & 0 & \circledast & 0 & 0 & 0 & \times & \times & \times & \times \\
0 & 0 & 0 & 0 & 0 & 0 & \circledast & \times & \times & \times
\end{array}\right] \\
&=: \left[\begin{array}{ccc|cc|c|ccc}
\times & \times & \times & \times & C_0 & & 0 & 0 & 0 \\
0 & \times & \times & \times & A_0 & C_1' & 0 & 0 & 0 \\
0 & \times & \times & 0 & & & \times & 0 & 0 \\
0 & 0 & D_0 & 0 & B_0 & & \times & \times & \times \\
0 & 0 & & 0 & & A_1' & \times & \times & \times \\
0 & 0 & \times & 0 & 0 \;\; 0 & & \times & \times & \times \\
0 & 0 & 0 & 0 & 0 \;\; 0 & B_1' & \times & \times & \times
\end{array}\right].
\end{aligned}
$$

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

$$
\begin{aligned}
\Phi_{-1}(z) &:= 0_3, \quad \Phi_0(z) := I_3 = [U_1 \;\; U_2 \;\; U_3], \\
\Phi_1(z) &:= [\phi_1(z) \;\; U_2 - e_{1,2}\phi_1(z) \;\; U_3 - e_{1,3}\phi_1(z)], \\
\Phi_2(z) &:= [\phi_2(z) \;\; U_2 - \sum_{i=1}^{2} e_{i,2}\phi_i(z) \;\; U_3 - \sum_{i=1}^{2} e_{i,3}\phi_i(z)], \\
\Phi_3(z) &:= [\phi_2(z) \;\; \phi_3(z) \;\; U_3 - \sum_{i=1}^{3} e_{i,3}\phi_i(z)], \\
\Phi_4(z) &:= [\phi_4(z) \;\; \phi_5(z) \;\; U_3 - \sum_{i=1}^{5} e_{i,3}\phi_i(z)], \\
\Phi_5(z) &:= [\phi_4(z) \;\; \phi_5(z) \;\; \phi_6(z)], \quad \Phi_6(z) := [\phi_7(z)],
\end{aligned}
$$

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

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

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

10. 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 (6.1). However, recurrence relation (6.2) can be rewritten using a decomposition of the matrix G.

THEOREM 10.1 (GENERALIZED BLOCK SCHUR PARAMETER DECOMPOSITION). *The unitary block Hessenberg matrix $G := Q^H \Lambda Q$ can be decomposed as*

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

ETNA
Kent State University
etna@mcs.kent.edu

with G_i having the form

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

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

$$G_j' =: \begin{bmatrix} \gamma_j & \Sigma_j \\ \sigma_j & \Gamma_j \end{bmatrix}, \tag{10.1}$$

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. We proceed 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

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

Note that $\sigma_{\pi_k'} = g_{k,\pi_k'}$ with k such that $\pi_k' = 1$. 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 4.1, we construct the blocks G_i', defined by (10.1), 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:

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

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 below the pivot elements.

ALGORITHM 10.1. *Transformation of the initial data matrix $\bar{D} := [\bar{E} \mid \bar{G}]$ 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)*:*

$$\bar{E} \leftarrow J^H \bar{E} \tag{10.2}$$

$$\bar{G} \leftarrow J^H \bar{G} \tag{10.3}$$

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

Note that (10.2) can be skipped if $\tau_i = n$. 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 10.1 changes as follows:

ALGORITHM 10.2. *Initialization*

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

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 $1 \leq \pi_i \leq n$ **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 at most $4n$ multiplications. The nontrivial part of $J^H H_{i-1}\bar{G}_i J$ is a unitary matrix of size at most $(2n+2)\times(2n+2)$. 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, as in the real case, is 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) $1\le\pi_k\le n$:

$$e_{k,\pi_k}Q_k = F'_{\pi_k}-\sum_{i=1}^{k-1}e_{i,\pi_k}Q_i \quad \text{(see (5.3))}. \tag{10.4}$$

b) $\pi_k-n=:\pi'_k>0$: We know that $\Lambda Q=QG_1G_2\ldots G_{m-r_m}$.
For $k=1,2,\ldots,m$, we define $Q_1^{(k)},Q_2^{(k)},\ldots,Q_k^{(k)}$ as

$$[Q_1^{(k)}\ Q_2^{(k)}\ldots Q_k^{(k)}\ Q_{k+1}^{(k)}\ Q_{k+2}^{(k)}\ldots Q_m^{(k)}]:=QG_1G_2\ldots G_{\pi'_j},$$

with

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

Note that if $\pi'_k>0$, 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

$$\begin{aligned} QG_1G_2\ldots G_{\pi'_k-1}G_{\pi'_k} &= [Q_1^{(k-1)}\ldots Q_{k-1}^{(k-1)}\ Q_k\ldots Q_m]G_{\pi'_k}\\ &= [Q_1^{(k)}\ldots Q_{k-1}^{(k)}\ Q_k^{(k)}\ Q_{k+1}\ldots Q_m]. \end{aligned} \tag{10.5}$$

If we partition the nontrivial $(\tau_j+1)\times(\tau_j+1)$ part G'_j of G_j (see (10.1)) using the block Schur parameters as

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

with σ_j a 1×1 block, we can rewrite (10.5) as

$$[Q_{\pi'_k}^{(k-1)}\ Q_{\pi'_k+1}^{(k-1)}\ldots Q_{k-1}^{(k-1)}\ Q_k]\begin{bmatrix}\gamma_{\pi'_k} & \Sigma_{\pi'_k}\\ \sigma_{\pi'_k} & \Gamma_{\pi'_k}\end{bmatrix}=[Q_{\pi'_k}^{(k)}\ Q_{\pi'_k+1}^{(k)}\ldots Q_{k-1}^{(k)}\ Q_k^{(k)}]. \tag{10.6}$$

Recalling from Theorem 10.1 that $\sigma_{\pi'_k}=g_{k,\pi'_k}$. Because the π'_k-th column of

$$\Lambda Q=QG_1G_2\ldots G_{m-\tau_m}$$

ETNA
Kent State University
etna@mcs.kent.edu

is equal to the π'_k-th column of $QG_1G_2\dots G_{\pi'_k}$, we get the following recurrence relation for Q_k by taking the first column of the left hand side of (10.6):

$$\sigma_{\pi'_k} Q_k = \Lambda Q_{\pi'_k} - [Q^{(k-1)}_{\pi'_k}\ Q^{(k-1)}_{\pi'_k+1}\dots Q^{(k-1)}_{k-1}]\gamma_{\pi'_k}. \tag{10.7}$$

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 (10.6), we get the following recurrence relation for the auxiliary columns $Q^{(k)}_j$, $j=\pi'_k+1,\dots,k$:

$$[Q^{(k)}_{\pi'_k+1}\dots Q^{(k)}_k] = [Q^{(k-1)}_{\pi'_k}\dots Q^{(k-1)}_{k-1}\ Q_k]\begin{bmatrix}\Sigma_{\pi'_k}\\ \Gamma_{\pi'_k}\end{bmatrix}. \tag{10.8}$$

The recurrence relations (10.4), (10.7) and (10.8) can be rewritten as recurrence relations with a limited number of terms. We proceed to rewrite this in terms of the orthonormal polynomial vectors ϕ_i. For each $k=1,2,\dots,m$, we start with

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

where

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

If $1\le\pi_k\le n$, we can use recurrence relation (10.4) to get

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

with

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

where

$$D_n := \begin{bmatrix} 0 & 0 & \dots & 0\\ 1 & 0 & \dots & 0\\ \vdots & \ddots & \ddots & \vdots\\ & & & 0\\ 0 & 0 & \dots & 1\end{bmatrix} \in \mathbb{C}^{n\times(n-1)}.$$

If $\pi_k - n =: \pi'_k > 0$, we can use recurrence relations (10.7) and (10.8) to get

$$[\phi_{k-n+1}\dots\phi_{k-1}|\phi_k|\phi^{(k)}_{k-\tau_k+1}\ \phi^{(k)}_{k-\tau_k+2}\dots\phi^{(k)}_k|S^{(k)}_{\tau_k+1}\dots S^{(k)}_n]$$

$$\leftarrow \quad [\phi_{k-n}\dots|\phi_{\pi'_k}|\dots\phi_{k-1}|\phi^{(k-1)}_{k-\tau_k}\dots\phi^{(k-1)}_{k-1}|S^{(k-1)}_{\tau_k+1}\dots S^{(k)}_n]\ T_k,$$

with

$$T_k := \left[\begin{array}{c|c|c|c} \multirow{3}{*}{D_n} & 0 & 0 & 0 \\ \cline{2-4} & \frac{z}{\sigma_{\pi'_k}} & \frac{z}{\sigma_{\pi'_k}}\Gamma_{\pi'_k} & \frac{-z}{\sigma_{\pi'_k}}[e_{k,\tau_k+1}\dots e_{k,n}] \\ \cline{2-4} & 0 & 0 & 0 \\ \hline 0 & \frac{-\gamma_{\pi'_k}}{\sigma_{\pi'_k}} & \Sigma_{\pi'_k} - \frac{\gamma_{\pi'_k}}{\sigma_{\pi'_k}}\Gamma_{\pi'_k} & \frac{\gamma_{\pi'_k}}{\sigma_{\pi'_k}}[e_{k,\tau_k+1}\dots e_{k,n}] \\ \hline 0 & 0 & 0 & I_{n-\tau_k} \end{array}\right].$$

Note that

$$\Sigma_{\pi'_k} - \gamma_{\pi'_k}\sigma^{-1}_{\pi'_k}\Gamma_{\pi'_k} = \Sigma^{-H}_{\pi'_k}$$

and

$$\sigma^{-1}_{\pi'_k}\Gamma_{\pi'_k} = -\gamma^H_{\pi'_k}\Sigma^{-H}_{\pi'_k}.$$

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 a diagram of this result.

EXAMPLE 10.1. Suppose the transformed data matrix has the following structure:

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

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

$$\begin{array}{ll}
A_0 := \begin{bmatrix} \frac{1}{e_{1,1}} & \frac{-e_{1,2}}{e_{1,1}} & \frac{-e_{1,3}}{e_{1,1}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & B_0 := 0_3,\\
C_0 := 0_3, & D_0 := 0_3,\\
A_1 := \begin{bmatrix} \frac{z}{\sigma_1} & \frac{-z}{\sigma_1}e_{2,2} & \frac{-z}{\sigma_1}e_{2,3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & B_1 := \begin{bmatrix} \frac{-\gamma_1}{\sigma_1} & \frac{-\gamma_1}{\sigma_1}e_{2,2} & \frac{-\gamma_1}{\sigma_1}e_{2,3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\\
C_1 := \begin{bmatrix} \frac{z}{\sigma_1}\Gamma_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & D_1 := \begin{bmatrix} \Sigma_1^{-H} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\\
A_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{e_{3,2}} & \frac{-e_{3,3}}{e_{3,2}} \\ 0 & 0 & 1 \end{bmatrix}, & B_2 := 0_3,\\
C_2 := \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{e_{3,2}} & 0 \\ 0 & 0 & 0 \end{bmatrix}, & D_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\\
A_3 := \begin{bmatrix} 0 & \frac{z}{\sigma_2} & \frac{-ze_{4,3}}{\sigma_2 e_{5,3}} \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & B_3 := \begin{bmatrix} 0 & \frac{-\gamma_2}{\sigma_2} & \frac{\gamma_2}{\sigma_2}\frac{e_{4,3}}{e_{5,3}} \\ 0 & 0 & 0 \end{bmatrix},\\
C_3 := \begin{bmatrix} \frac{z}{\sigma_2}\Gamma_2 & \frac{-ze_{4,3}}{\sigma_2 e_{5,3}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, & D_3 := \begin{bmatrix} \Sigma_2^{-H} & \frac{\gamma_2 e_{4,3}}{\sigma_2 e_{5,3}} \\ 0 & 0 \end{bmatrix},\\
A_4 := z\ \sigma_{3,5}^{-1}, & B_4 := -\gamma_{3,5}\sigma_{3,5}^{-1},\\
C_4 := -\gamma_{3,5}\Sigma_{3,5}^{-H}, & D_4 := \Sigma_{3,5}^{-H},
\end{array}$$

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.

11. Application: linearized vector rational approximation. In this section, we generalize the discrete linearized least-squares rational approximation of [16, 17] to the vector case. The definitions can be easily extended to the vector case as follows. Let us assume that the function values are given in the form E_i/f_i for the abscissae $z_i, i = 1, 2, \ldots, m$, where $E_i \in \mathbb{C}^{(n-1)\times 1}$ and f_i and z_i are all complex numbers. In the sequel, we make no distinction between the vector rational form $N(z)/d(z)$ and the polynomial vector $P =: [P_1, P_2, \ldots, P_n]^T$ with $N := [P_1, P_2, \ldots, P_{n-1}]^T$ and $d := P_n$. If we define the $\vec{\tau}$-degree of a polynomial vector P with $\vec{\tau} \in \mathbb{Z}^n$ as $\vec{\tau}\text{-deg}\, P(z) := \max\{\deg P_i(z) - \tau_i\}$ ($\deg 0 = -1$), then in the rational interpolation problem, one wants to describe all vector rational forms $N(z)/d(z)$ of minimal $\vec{\tau}$-degree which satisfy the interpolation conditions

$$\frac{N(z_i)}{d(z_i)} = \frac{E_i}{f_i}, \qquad i = 1, 2, \ldots, m. \tag{11.1}$$

In [15], a parametrization was given and an efficient algorithm to solve this problem. When the data are corrupted with noise, one does not want the interpolation conditions to be satisfied exactly.

DEFINITION 11.1 (PROPER VECTOR RATIONAL APPROXIMATION PROBLEM). *Given the data points z_i with corresponding estimated function values $E_i/f_i, i = 1, 2, \ldots, m$, given $\vec{\tau}$, the $\vec{\tau}$-degree α and the weights $w_i^{(p)} > 0, i = 1, 2, \ldots, m$, we look for a vector rational form $N(z)/d(z)$ of $\vec{\tau}$-degree $\leq \alpha$ satisfying the following least squares approximation criterion:*

$$\textit{minimize } \operatorname{dist}^2_{(p)}(N, d) := \sum_{i=1}^{m} w_i^{(p)} \|R_i^{(p)}\|_2^2, \tag{11.2}$$

with $R_i^{(p)} := E_i/f_i - N(z_i)/d(z_i)$, $\|R_i^{(p)}\|_2^2 := (R_i^{(p)})^H R_i^{(p)}$ and where $\operatorname{dist}_{(p)}(N, d)$ denotes the l_2-distance between the rational function $N(z)/d(z)$ and the data.

The proper vector rational approximation problem is a non-linear least squares problem which can only be solved in an iterative way. Therefore, we rather minimize the norm of the linearized residual vector with components

$$R_i := E_i d(z_i) - f_i N(z_i), \qquad i = 1, 2, \ldots, m. \tag{11.3}$$

We shall fix the $\vec{\tau}$-degree of the approximant $N(z)/d(z)$ to be α where usually $\alpha \ll m$. We also normalize the approximant in the following sense. Suppose $P_i(z) =: P_{i,0} + P_{i,1}z + \cdots + P_{i,\alpha+\tau_i} z^\alpha$, then we require $P_{i,\alpha+\tau_i} = 1$ if $P_{j,\alpha+\tau_j} = 0$, $j = i+1, i+2, \ldots, n$. Thus, we shall solve the following vector rational approximation problem.

DEFINITION 11.2 (LINEARIZED VECTOR RATIONAL APPROXIMATION PROBLEM). *Given the data points z_i with corresponding estimated function values $E_i/f_i, i = 1, 2, \ldots, m$, given $\vec{\tau}$, the $\vec{\tau}$-degree α and the weights $w_i > 0, i = 1, 2, \ldots, m$, we look for the normalized rational form $N(z)/d(z)$ of $\vec{\tau}$-degree α satisfying the following least squares approximation criterion:*

$$\textit{minimize } \operatorname{dist}^2(N, d) := \sum_{i=1}^{m} w_i \|R_i\|_2^2, \tag{11.4}$$

with $R_i := E_i d(z_i) - f_i N(z_i)$, $\|R_i\|_2^2 = {R_i}^H R_i$ and where $\operatorname{dist}(N, d)$ denotes the distance between the vector rational form $(N(z), d(z))$ and the data.

Note that the function values E_i/f_i can be replaced by $k_iE_i/(k_if_i)$ (with $k_i \neq 0$). This yields a different value of the residual R_i. Solving the linearized rational approximation problem with k_iE_i, k_if_i instead of E_i and f_i, is equivalent to solving the problem with the original E_i, f_i but with the weights $w_i|k_i|^2$ instead of w_i.

The solution of the linearized problem can be used to obtain a solution of the proper problem as follows. Suppose we know the values $d^{(p)}(z_i), i = 1, 2, \dots, m$ with $N^{(p)}/d^{(p)}$ a solution of the proper problem. If we solve the linearized problem with weights

$$w_i = \frac{w_i^{(p)}}{|f_i d^{(p)}(z_i)|^2}, \qquad i = 1, 2, \dots, m, \tag{11.5}$$

we get $(N^{(p)}, d^{(p)})$. However, in practice, we do not know the values $d^{(p)}(z_i)$. In this case we can estimate these values, compute the solution of the linearized problem, take the denominator of this solution as a new estimation of the final $d^{(p)}$, and so on. This algorithm was proposed by Loeb for the l_∞ norm and by Wittmeyer for the l_2 norm [3]. In Example 11.1 we shall show the influence of executing one iteration step of this algorithm. Of course one could also use the solution of the linearized problem as a starting value for other iterative schemes.

The linearized vector rational approximation problem can be formulated as a discrete least squares approximation problem with polynomial vectors where each point z_i is taken with $(n-1)$ different weight vectors, the rows of the $(n-1) \times n$ matrix

$$\sqrt{w_i}[If_i \quad -E_i].$$

The degree vector $\Delta^\star$ can be taken equal to $\vec{\tau}$.

EXAMPLE 11.1. The points z_i are 30 equidistant points in the interval $[-\pi/2 + 0.01, \pi/2 + 0.01]$. The function values are taken as follows:

$$f_i = 1, \qquad E_i = [\tan(z_i), \sin(z_i)]^T.$$

The weights w_i and $w_i^{(p)}$ are all taken equal to 1. We allow the degree of the numerator to be 2 higher than the degree of the denominator, i.e. $\vec{\tau} = (2, 2, 0)$. All computations were done on a DEC workstation using MATLAB 4.1 with machine epsilon $eps = 2.2204e{-}16$. Table 11.1 shows the degree vectors $\Delta^{(k)}$, the degree indices ν_k, the pivot indices π_k and the norm of the solution of the discrete least squares approximation for the initial values of k. The absolute errors of each of the two components of the vector rational approximant for $k = 19$ is given in Figure 11.1. After one iteration step of the Loeb-Wittmeyer algorithm, we get the smaller errors of Figure 11.2 with a more equi-oscillating character.

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 by 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. As an application, we have considered the vector rational approximation problem in a linearized least-squares sense. Future work will show how to compute matrix rational interpolants and matrix rational approximants in a linearized least squares sense based on these orthonormal polynomial vectors.

k	$\Delta^{(k)}$	ν_k	π_k	$\|P^{(k)}\|$
1	(0,-1,-1)	1	1	5.4772e+00
2	(0,0,-1)	2	2	5.4772e+00
3	(1,0,-1)	1	4	5.1030e+00
4	(1,1,-1)	2	5	5.1030e+00
5	(2,1,-1)	1	6	4.2454e+00
6	(2,2,-1)	2	7	4.2454e+00
7	(2,2,0)	3	3	1.2223e+02
8	(3,2,0)	1	8	2.5927e+00
9	(3,3,0)	2	9	3.4585e+00
10	(3,3,1)	3	10	1.6908e+02
11	(4,3,1)	1	11	1.9535e+00
12	(4,4,1)	2	12	2.7890e+00
13	(4,4,2)	3	13	3.4205e-01
14	(5,4,2)	1	14	1.4593e+00
15	(5,5,2)	2	15	7.0727e-02
16	(5,5,3)	3	16	2.6297e-01
17	(6,5,3)	1	17	1.0807e+00
18	(6,6,3)	2	18	5.6111e-02
19	(6,6,4)	3	19	8.0443e-03
20	(7,6,4)	1	20	3.3817e-01
21	(7,7,4)	2	21	1.4988e-03
22	(7,7,5)	3	22	5.9541e-03
23	(8,7,5)	1	23	2.5137e-01
24	(8,8,5)	2	24	1.0902e-03
25	(8,8,6)	3	25	2.5033e-03

TABLE 11.1
results for example 11.1

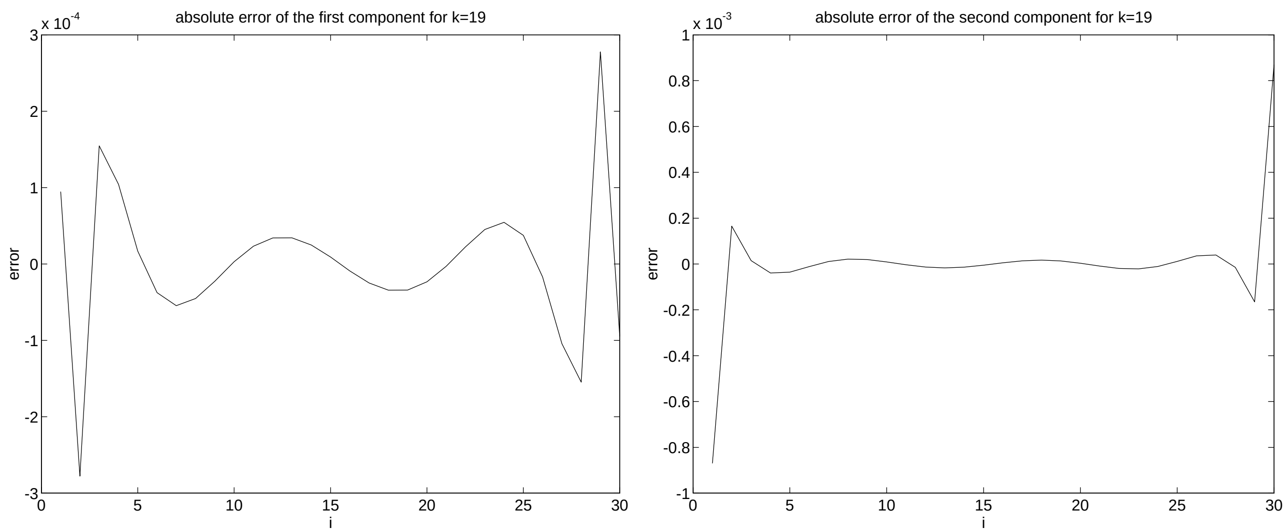

FIG. 11.1. *absolute error*

REFERENCES

[1] G. AMMAR AND W. GRAGG, $O(n^2)$ *reduction algorithms for the construction of a band matrix from spectral data*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 426–431.

[2] G. AMMAR, W. GRAGG, AND L. REICHEL, *Constructing a unitary Hessenberg matrix from spectral data*, in Numerical linear algebra, digital signal processing and parallel algorithms, G. Golub and P. Van Dooren, eds., vol. 70 of NATO-ASI Series, F: Computer and Systems

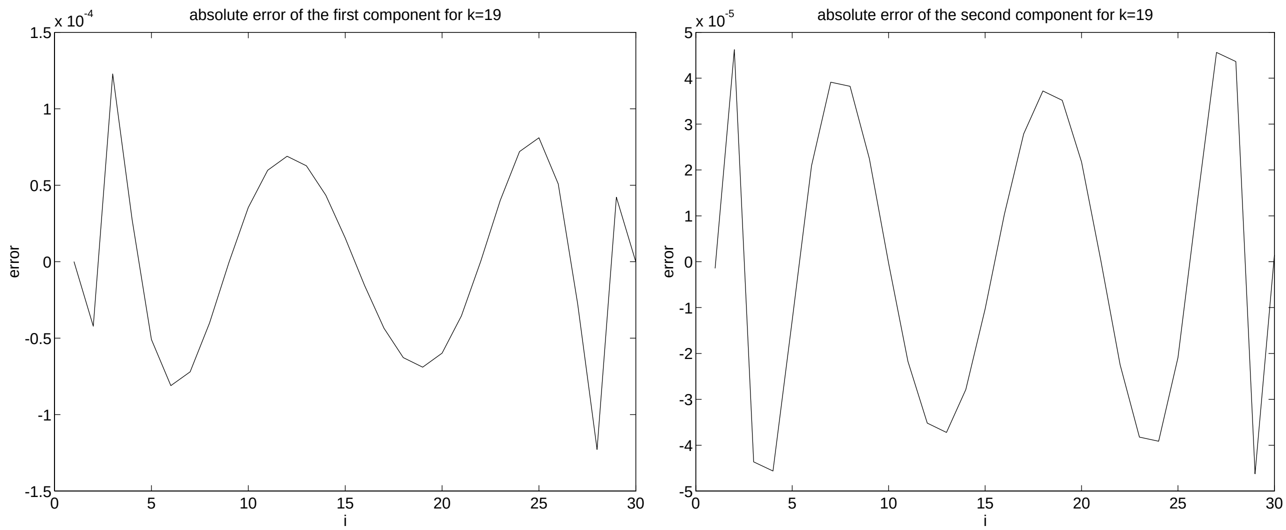

FIG. 11.2. *absolute error after one iteration step*

Sciences, Berlin, 1991, Springer-Verlag, pp. 385–395.

[3] I. Barrodale and J. Mason, *Two simple algorithms for discrete rational approximation*, Math. Comp., 24 (1970), pp. 877–891.

[4] D. Boley and G. Golub, *A survey of matrix inverse eigenvalue problems*, Inverse Problems, 3 (1987), pp. 595–622.

[5] A. Bultheel and M. Van Barel, *Vector orthogonal polynomials and least squares approximation*, SIAM J. Matrix Anal. Appl., (1994), to appear.

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

[7] V. Dubovoj, B. Fritzsche, and B. Kirstein, *Matricial version of the classical Schur problem*, vol. 129 of Teubner-Texte Math., Teubner Verlagsgesellschaft, Stuttgart, Leipzig, 1992.

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

[9] J. Geronimo, *Matrix orthogonal polynomials on the unit circle*, J. Math. Phys., 22 (1981), pp. 1359–1365.

[10] ———, *Scattering theory and matrix orthogonal polynomials on the real line*, Circuits Systems Signal Process., 1 (1982), pp. 471–495.

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

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

[13] L. Reichel, G. Ammar, and W. Gragg, *Discrete least squares approximation by trigonometric polynomials*, Math. Comp., 57 (1991), pp. 273–289.

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

[15] M. Van Barel and A. Bultheel, *A new approach to the rational interpolation problem: the vector case.*, J. Comput. Appl. Math., 33 (1990), pp. 331–346.

[16] ———, *A parallel algorithm for discrete least squares rational approximation*, Numer. Math., 63 (1992), pp. 99–121.

[17] ———, *Discrete linearized least-squares rational approximation on the unit circle*, J. Comput. Appl. Math., 50 (1994), pp. 545–563.