Discrete linearized least squares rational approximation on the unit circle

Marc Van Barel and Adhemar Bultheel
Department of Computer Science
K.U. Leuven

February 23, 1999

Abstract

We already generalized the Rutishauser-Gragg-Harrod-Reichel algorithm for discrete least squares polynomial approximation on the real axis to the rational case. In this paper, a new method for discrete least squares linearized rational approximation on the unit circle is presented. It generalizes the algorithm of Reichel-Ammar-Gragg for discrete least squares polynomial approximation on the unit circle to the rational case. The algorithm is fast in the sense that it requires order $m\alpha$ computation time where m is the number of data points and α is the degree of the approximant. We describe how this algorithm can be implemented in parallel. Examples illustrate the numerical behavior of the algorithm.

Keywords : rational interpolation, rational approximation, linearized least squares, (block) unitary Hessenberg, (block) Schur parameters, (block) orthonormal polynomials, parallel

AMS(MOS) Classification : Primary : 41A20, Secondary : 65D05, 65D15, 30E10

Discrete linearized least squares rational approximation on the unit circle

Marc Van Barel and Adhemar Bultheel
Department of Computer Science
K.U. Leuven

February 23, 1999

Abstract

We already generalized the Rutishauser-Gragg-Harrod-Reichel algorithm for discrete least squares polynomial approximation on the real axis to the rational case. In this paper, a new method for discrete least squares linearized rational approximation on the unit circle is presented. It generalizes the algorithm of Reichel-Ammar-Gragg for discrete least squares polynomial approximation on the unit circle to the rational case. The algorithm is fast in the sense that it requires order $m\alpha$ computation time where m is the number of data points and α is the degree of the approximant. We describe how this algorithm can be implemented in parallel. Examples illustrate the numerical behavior of the algorithm.

1 Introduction

In our previous publications [14, 15], we investigated the rational interpolation problem not only for scalar function values but also for the vector case. The standpoint we took was a theoretical one. We found a parametrization of all solutions of the rational interpolation problem and fast algorithms to compute this parametrization. Fast means of order m^2 floating point operations (flops) if there are m interpolation data. However, the algorithms which we designed are not numerically stable. This limits their applicability to theoretical investigation of the rational interpolation problem or to those cases where there are no round-off errors, e.g. when the computations are done within a finite field.

In our paper [16], we replaced the interpolation conditions by a discrete least squares criterion and described an algorithm to solve this problem when all interpolation points are taken on the real axis. It generalizes the algorithm of Reichel [9] for the polynomial case. The latter is inspired by the Rutishauser-Gragg-Harrod algorithm [12, 8, 1] for the computation of Jacobi matrices. A similar way of reasoning was followed in [4, 6]. In this paper, we construct an algorithm solving the complementary problem, i.e. when all interpolation points are taken on the unit circle. This algorithm generalizes the algorithm given by Reichel, Ammar and Gragg [10]. The latter is inspired by the inverse unitary QR algorithm for computing unitary Hessenberg matrices [2]. It will turn out that our algorithm is fast, more specifically it is of order $m\alpha$ flops where α indicates the degree of the rational approximant. Implementing this algorithm in a pipelined fashion on α processors working in parallel reduces the computational time to order m.

2 The theory behind the algorithm

Let us assume that the function values are given in the form f_i/e_i for the abscissae $z_i, i = 1, 2, \ldots, m$, where f_i, e_i and z_i are all complex numbers. If we define the degree of a rational form $n(z)/d(z)$ with $(n, d) \neq (0, 0)$ as $\deg(n(z), d(z)) = \max\{\deg n(z), \deg d(z)\}$ $(\deg 0 = -1)$, then in the rational interpolation problem, one wants to describe all polynomial couples $(n(z), d(z))$ of minimal degree which satisfy the interpolation conditions

$$\frac{n(z_i)}{d(z_i)} = \frac{f_i}{e_i}, \qquad i = 1, 2, \ldots, m.$$

In [14], 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 2.1 (proper rational approximation problem) *Given the data points z_i with corresponding estimated function values $f_i/e_i, i = 1, 2, \ldots, m$, given the degree α and the weights $w_i^{(p)} > 0, i = 1, 2, \ldots, m$,*

we look for a rational form $n(z)/d(z)$ of degree $\leq \alpha$ satisfying the following least squares approximation criterion:

$$\textit{minimize } \operatorname{dist}_{(p)}^2(n,d) = \sum_{i=1}^{m} w_i^{(p)} |r_i^{(p)}|^2,$$

with $r_i^{(p)} = f_i/e_i - n(z_i)/d(z_i)$, $|r_i^{(p)}|^2 = \overline{r_i^{(p)}} r_i^{(p)}$ 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 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

$$r_i = f_i d(z_i) - e_i n(z_i), \qquad i = 1, 2, \ldots, m.$$

We shall fix the degree of the approximant $n(z)/d(z)$ to be $\alpha = \max\{\deg n(z), \deg d(z)\}$ where usually $\alpha \ll m$. We also normalize the approximant in the following sense. Suppose $n(z) = n_0 + n_1 z + \cdots + n_\alpha z^\alpha$ and $d(z) = d_0 + d_1 z + \cdots + d_\alpha z^\alpha$, then we require $n_\alpha = 1$ if $\deg n(z) = \alpha$ and $d_\alpha = 1$ if $\deg n(z) < \alpha$ (and $\deg d(z) = \alpha$). Note that we have given preference to the numerator polynomial in the normalization procedure. However, by switching the role of f_i and e_i and of $n(z)$ and $d(z)$, we get the symmetric problem giving preference to $d(z)$ in the normalization procedure. Thus in this paper, we shall solve the following rational approximation problem.

Definition 2.2 (linearized rational approximation problem) *Given the data points z_i with corresponding estimated function values $f_i/e_i, i = 1, 2, \ldots, m$, given the 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 degree α satisfying the following least squares approximation criterion:*

$$\textit{minimize } \operatorname{dist}^2(n,d) = \sum_{i=1}^{m} w_i |r_i|^2,$$

with $r_i = f_i d(z_i) - e_i n(z_i)$, $|r_i|^2 = \overline{r_i} r_i$ and where $\operatorname{dist}(n,d)$ denotes the distance between the rational form $(n(z), d(z))$ and the data.

Note that the function values f_i/e_i can be replaced by $k_i f_i/(k_i e_i)$ (with $k_i \neq 0$). This yields a different value of the residual r_i. Solving the linearized rational approximation problem with $k_i f_i$, $k_i e_i$ instead of f_i and e_i, is equivalent to solving the problem with the original f_i, e_i but with the weights $w_i |k_i|^2$ instead of w_i. Choosing k_i, $i = 1, 2, \ldots, m$ can be done in several ways, e.g. one could normalize the data such that $|f_i|^2 + |e_i|^2 = 1$ or such that $\max\{|f_i|, |e_i|\} = 1$, $i = 1, 2, \ldots, m$.

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

$$w_i = \frac{w_i^{(p)}}{|e_i d^{(p)}(z_i)|^2}, \qquad i = 1, 2, \ldots, m$$

we get $(n^{(p)}, d^{(p)})$. However, in practice, we do not know the values $d^{(p)}(z_i)$. In this case we can make an estimation of 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 Examples 7.2 and 7.3 of Section 7, 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.

Let us now look how the linearized problem can be solved in an efficient way. The linear least squares problem can be reformulated as a block problem, i.e. we should minimize

$$\sum_{i=1}^{m} \left[\overline{d(z_i)} \;\; \overline{n(z_i)} \right] v_i \begin{bmatrix} d(z_i) \\ n(z_i) \end{bmatrix}$$

with

$$v_i = \begin{bmatrix} \overline{f_i'} \\ -\overline{e_i'} \end{bmatrix} \begin{bmatrix} f_i' & -e_i' \end{bmatrix} \qquad \text{and} \qquad e_i' = \sqrt{w_i} e_i \qquad \text{and} \qquad f_i' = \sqrt{w_i} f_i.$$

Denoting the approximant as $a(z) = [d(z)\ n(z)]^T$, this can be formulated as

$$\sum_{i=1}^{m} w_i |r_i|^2 = ||a(z)||_v^2$$

with

$$||a(z)||_v^2 = \langle a(z), a(z)\rangle_v \qquad \text{and} \qquad \langle a(z), b(z)\rangle_v = \sum_{i=1}^{m} \overline{a(z_i)}^T v_i b(z_i)$$

for any $a(z)$, and $b(z)$ polynomial vectors in $\mathbf{C}[z]^2$. We also need a matrix version. So, if $A(z), B(z) \in \mathbf{C}[z]^{2\times 2}$, we define

$$\langle A(z), B(z)\rangle_v = \sum_{i=1}^{m} \overline{A(z_i)}^T v_i B(z_i).$$

Now suppose that we can construct some orthonormal polynomials $\{\phi_0, \phi_1, \ldots\}$ in $\mathbf{C}[z]^{2\times 2}$, such that $\phi_k(z) = \phi_{kk} z^k + \cdots + \phi_{k0}$ with ϕ_{kk} regular and upper triangular and such that

$$\langle \phi_i(z), \phi_j(z)\rangle_v = \delta_{ij} I_2$$

with I_2 the 2×2 unit matrix. Then we can express $a(z)$ as a linear combination

$$a(z) = \sum_{i=0}^{\alpha} \phi_i(z) a_i, \qquad a_i \in \mathbf{C}^2.$$

Thus,

$$\begin{aligned} ||a(z)||_v^2 &= \sum_{i=1}^{m} \overline{a(z_i)}^T v_i a(z_i) = \sum_{i=1}^{m} (\sum_{k=0}^{\alpha} \overline{a_k}^T \overline{\phi_k(z_i)}^T) v_i (\sum_{l=0}^{\alpha} \phi_l(z_i) a_l) \\ &= \sum_{k,l=0}^{\alpha} \overline{a_k}^T (\sum_{i=1}^{m} \overline{\phi_k(z_i)}^T v_i \phi_l(z_i)) a_l = \sum_{k,l=0}^{\alpha} \overline{a_k}^T \langle \phi_k, \phi_l\rangle_v a_l = \sum_{k=0}^{\alpha} \overline{a_k}^T a_k \\ &= \sum_{k=0}^{\alpha} ||a_k||^2. \end{aligned}$$

Because a_α can not be zero, but all a_k, $k = 0, 1, \ldots, \alpha - 1$ can be chosen as zero, we find that for the minimizing $a(z)$

$$a(z) = \sum_{k=0}^{\alpha} \phi_k(z) a_k = \phi_\alpha(z) a_\alpha,$$

where a_α is chosen to minimize

$$||a(z)||_v^2 = \overline{a_\alpha}^T a_\alpha = |\rho_\alpha|^2 + |\tau_\alpha|^2 \qquad \text{with} \qquad a_\alpha = [\rho_\alpha\ \tau_\alpha]^T.$$

Given the normalizing conditions for the approximant and the leading coefficient of $\phi_\alpha(z)$, this is an easy problem to solve.

Our first task will be to construct the orthogonal block polynomials ϕ_j. We use a generalization of the method introduced by Reichel, Ammar and Gragg [10] and Reichel [9] to compute a polynomial approximation in the least squares sense.

3 The algorithm

The algorithm starts by applying some unitary similarity transformations on the data which are first arranged in an array as follows

$$M = \left[\begin{array}{c|c} 0 & \begin{bmatrix} f' & -e' \end{bmatrix}^H \\ \hline \begin{bmatrix} f' & -e' \end{bmatrix} & Z \end{array}\right]$$

where we used the notation

$$\begin{bmatrix} f' & -e' \end{bmatrix} = \begin{bmatrix} f'_1 & f'_2 & \cdots & f'_m \\ -e'_1 & -e'_2 & \cdots & -e'_m \end{bmatrix}^T, \qquad Z = \operatorname{diag}(z_1, z_2, \ldots, z_m).$$

We have a double column in the SW quadrant, which is reflected as complex conjugate transposed rows in the NE quadrant, and in the SE quadrant we originally find the diagonal matrix of abscissae. We now apply unitary similarity transformations to the left and right of this initial matrix M to transform it into $Q'^H M Q'$, a block upper Hessenberg matrix (the blocks are two by two), where $Q' = I_2 \oplus Q$. Therefore, if we take all abscissae z_i on the unit circle, i.e. $|z_i| = 1$, $i = 1, 2, \ldots, m$, $Q^H Z Q$ will be unitary (block upper Hessenberg). When the data points lie on the unit circle, Reichel, Ammar and Gragg [10] solve the least squares polynomial approximation problem using a Schur parametrization of the unitary Hessenberg matrix that plays the central role in the theory.

Let us return to the computational scheme and explain how the transformation $Q'^H M Q'$ is actually performed using order m^2 computational work. Therefore, we need the following properties of unitary block Hessenberg matrices.

Theorem 3.1 (block Schur parametrization) *Every unitary block Hessenberg matrix H of dimension m can be represented as*

$$H = G_1 G_2 G_3 \cdots G_{m'-1}$$

with $m' = \lfloor (m+1)/2 \rfloor$, the greatest natural number smaller than or equal to $(m+1)/2$. Each G_k is a unitary matrix of the form

$$G_k = \begin{bmatrix} I_{2(k-1)} & & & \\ & -\gamma_k & \hat{\sigma}_k & \\ & \sigma_k & \hat{\gamma}_k & \\ & & & I_{m-2(k+1)} \end{bmatrix}, \qquad k = 1, 2, \ldots, m' - 1.$$

Note that I_k is the $(k \times k)$ unit matrix. For $k \leq 0$, this represents the empty matrix. The matrices γ_k, σ_k, $\hat{\gamma}_k$ and $\hat{\sigma}_k$ (block Schur parameters) are (2×2) except when m is odd and $k = m' - 1$. In this case $\gamma_{m'-1}$ is (2×2), $\sigma_{m'-1}$ is (1×2), $\hat{\sigma}_{m'-1}$ is (2×1) and $\hat{\gamma}_{m'-1}$ is (1×1). The matrices σ_k, $k = 1, 2, \ldots, m' - 1$ are the subdiagonal blocks of the unitary block Hessenberg matrix H.

Proof. Each unitary block Hessenberg matrix H can be written as

$$H = \begin{bmatrix} -\gamma_1 & \times \\ \sigma_1 & \times \\ 0 & \times \end{bmatrix}.$$

Because $H^H H = I_m$, we have that $\gamma_1^H \gamma_1 + \sigma_1^H \sigma_1 = I_2$. Hence, there exist matrices $\hat{\gamma}_1$ and $\hat{\sigma}_1$ such that

$$G_1 = \begin{bmatrix} -\gamma_1 & \hat{\sigma}_1 & \\ \sigma_1 & \hat{\gamma}_1 & \\ & & I_{m-4} \end{bmatrix}$$

is unitary. Therefore, $G_1^H H$ is unitary and equal to

$$G_1^H H = \begin{bmatrix} I_2 & X \\ 0 & H' \end{bmatrix}.$$

Because $H H^H = I_m$, $X X^H = 0$ or $X = 0$. Hence, we can continue the same way of reasoning considering the smaller unitary block Hessenberg matrix H' to get G_2 and so on. Note that the subdiagonal blocks of H' are the same as those of H skipping the first one. □

In the sequel, we also need the following theorem.

Theorem 3.2 *Given the (1×2) matrix A and the (2×2) matrix B, there always exists a unitary matrix T such that*

$$T^H \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} C \\ 0_{1\times 2} \end{bmatrix}$$

with C a (2×2) upper triangular matrix.

Proof. A possible choice for T follows by taking the QR factorization of the matrix $\left[A^H \ B^H\right]^H$. □

We could transform the initial array M into a unitarily similar block Hessenberg matrix using Givens reflections (or rotations) or Householder transformations. However, this would require $O(m^3)$ computational work. Because $H = Q^H Z Q$ is unitary block Hessenberg, we can parametrize this matrix using block Schur parameters. We now construct a recursive algorithm where we transform these block Schur parameters in each step adding one new data point $\lambda = z_{m+1}, f_{m+1}, e_{m+1}, w_{m+1}$. After m steps, the initial data matrix M is transformed into

$$Q'^H M Q' = \left[\begin{array}{c|cc} 0 & \sigma_0^H & 0 \\ \hline \sigma_0 & & \\ 0 & \multicolumn{2}{c}{H} \end{array}\right]$$

with $H = Q^H Z Q$ a unitary block Hessenberg matrix with upper triangular subdiagonal blocks $\sigma_1, \sigma_2, \ldots, \sigma_{m'-1}$ (as well as σ_0). In step $m+1$, we add the $(m+1)$-st data point

$$\left[\begin{array}{c|cc|c} 0 & \sigma_0^H & 0 & \pi_0^H \\ \hline \sigma_0 & & & \\ 0 & \multicolumn{2}{c|}{H} & 0 \\ \hline \pi_0 & 0 & & \lambda \end{array}\right]$$

with $\pi_0 = \left[f'_{m+1} \ -e'_{m+1}\right]$ which is unitarily similar to

$$\left[\begin{array}{c|c|cc} 0 & \pi_0^H & \sigma_0^H & 0 \\ \hline \pi_0 & \lambda & \multicolumn{2}{c}{0} \\ \hline \sigma_0 & 0 & \multicolumn{2}{c}{H} \\ 0 & & & \end{array}\right].$$

Using Theorem 3.2, we derive a unitary matrix T_0

$$T_0 = \begin{bmatrix} -\mu_0 & \hat{\nu}_0 \\ \nu_0 & \hat{\mu}_0 \end{bmatrix} \qquad \text{such that} \qquad T_0^H \begin{bmatrix} \pi_0 \\ \sigma_0 \end{bmatrix} = \begin{bmatrix} \sigma'_0 \\ 0 \end{bmatrix}$$

with σ'_0 upper triangular. Using a block Schur parametrization for $H = G_1 G_2 \cdots G_{m'-1}$, we can write

$$\begin{bmatrix} T_0^H & 0 \\ 0 & I_{m-2} \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & H \end{bmatrix} = \begin{bmatrix} T_0^H & 0 \\ 0 & I_{m-2} \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & I_m \end{bmatrix} \prod_{i=1}^{m'-1} \begin{bmatrix} 1 & 0 \\ 0 & G_i \end{bmatrix}$$

which is again unitary block Hessenberg. Hence, $\begin{bmatrix} Z & \\ & \lambda \end{bmatrix}$ is unitarily similar to

$$\begin{bmatrix} T_0^H & 0 \\ 0 & I_{m-2} \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & H \end{bmatrix} \begin{bmatrix} T_0 & 0 \\ 0 & I_{m-2} \end{bmatrix}$$

which is unitary block Hessenberg except for the principal (3×3) block submatrix. With the notation

$$\begin{bmatrix} -\alpha_0^H & \beta_0^H \\ \hat{\beta}_0^H & \hat{\alpha}_0^H \end{bmatrix} = \begin{bmatrix} -\mu_0^H & \nu_0^H \\ \hat{\nu}_0^H & \hat{\mu}_0^H \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & I_2 \end{bmatrix}$$

we can compute this (3×3) block or (5×5) principal submatrix as

$$\begin{bmatrix} -\alpha_0^H & \beta_0^H & \\ \hat{\beta}_0^H & \hat{\alpha}_0^H & \\ & & I_2 \end{bmatrix} \begin{bmatrix} 1 & & \\ & -\gamma_1 & \hat{\sigma}_1 \\ & \sigma_1 & \hat{\gamma}_1 \end{bmatrix} \begin{bmatrix} -\mu_0 & \hat{\nu}_0 & \\ \nu_0 & \hat{\mu}_0 & \\ & & I_2 \end{bmatrix} =$$
$$\begin{bmatrix} \alpha_0^H \mu_0 - \beta_0^H \gamma_1 \nu_0 & -\alpha_0^H \hat{\nu}_0 - \beta_0^H \gamma_1 \hat{\mu}_0 & \beta_0^H \hat{\sigma}_1 \\ -\hat{\beta}_0^H \mu_0 - \hat{\alpha}_0^H \gamma_1 \nu_0 & \hat{\beta}_0^H \hat{\nu}_0^H - \hat{\alpha}_0^H \gamma_1 \hat{\mu}_0 & \hat{\alpha}_0^H \hat{\sigma}_1 \\ \sigma_1 \nu_0 & \sigma_1 \hat{\mu}_0 & \hat{\gamma}_1 \end{bmatrix}.$$

Using Theorem 3.2, we can always find a unitary matrix T_1

$$T_1 = \begin{bmatrix} -\mu_1 & \hat{\nu}_1 \\ \nu_1 & \hat{\mu}_1 \end{bmatrix} \qquad \text{such that} \qquad T_1^H \begin{bmatrix} \pi_1 \\ \sigma_1 \nu_0 \end{bmatrix} = \begin{bmatrix} \sigma'_1 \\ 0 \end{bmatrix}$$

with σ_1' upper triangular and $\pi_1 = -\hat{\beta}_0^H \mu_0 - \hat{\alpha}_0^H \gamma_1 \nu_0$. Multiplying the (5×5) principal submatrix to the left by $I_2 \oplus T_1^H$, we get a unitary block Hessenberg matrix which can be written with $-\gamma_1' = \alpha_0^H \mu_0 - \beta_0^H \gamma_1 \nu_0$ and the σ_1' from above using block Schur parameters as

$$\begin{bmatrix} -\gamma_1' & \hat{\sigma}_1' & \\ \sigma_1' & \hat{\gamma}_1' & \\ & & 1 \end{bmatrix} \begin{bmatrix} I_2 & & \\ & -\alpha_1^H & \beta_1^H \\ & \hat{\beta}_1^H & \hat{\alpha}_1^H \end{bmatrix}.$$

To get a unitarily similar matrix, we have to multiply the unitary block Hessenberg matrix

$$\begin{bmatrix} -\gamma_1' & \hat{\sigma}_1' & & \\ \sigma_1' & \hat{\gamma}_1' & & \\ & & I_{m-3} & \end{bmatrix} \begin{bmatrix} I_2 & & & \\ & -\alpha_1^H & \beta_1^H & \\ & \hat{\beta}_1^H & \hat{\alpha}_1^H & \\ & & & I_{m-4} \end{bmatrix} \prod_{i=2}^{m'-1} \begin{bmatrix} 1 & 0 \\ 0 & G_i \end{bmatrix}$$

to the right by $I_2 \oplus T_1 \oplus I_{m-4}$. Hence, we can repeat the same reasoning for $T_1, T_2, T_3, \ldots$ as for T_0. Before the final step of the algorithm is executed, we have the following unitarily similar $(m+1) \times (m+1)$ matrix

$$G_1' G_2' \cdots G_{m'-2}' \begin{bmatrix} I_{2(m'-2)} & & & \\ & -\alpha_{m'-2}^H & \beta_{m'-2}^H & \\ & \hat{\beta}_{m'-2}^H & \hat{\alpha}_{m'-2}^H & \\ & & & I_\delta \end{bmatrix} \begin{bmatrix} 1 & & \\ & G_{m'-1} & \\ & & \end{bmatrix} \begin{bmatrix} I_{2(m'-2)} & & & \\ & & -\mu_{m'-2} & \hat{\nu}_{m'-2} \\ & & \nu_{m'-2} & \hat{\mu}_{m'-2} \\ & & & I_\delta \end{bmatrix}$$

with $\delta = 1$ if m is odd, and $\delta = 2$ if m is even. The unitary matrices G_k' are built up using the prime block Schur parameters. The final step transforms the last $(3+\delta) \times (3+\delta)$ block

$$B = \begin{bmatrix} -\alpha_{m'-2}^H & \beta_{m'-2}^H & \\ \hat{\beta}_{m'-2}^H & \hat{\alpha}_{m'-2}^H & \\ & & I_\delta \end{bmatrix} \begin{bmatrix} 1 & & \\ & -\gamma_{m'-1} & \hat{\sigma}_{m'-1} \\ & \sigma_{m'-1} & \hat{\gamma}_{m'-1} \end{bmatrix} \begin{bmatrix} -\mu_{m'-2} & \hat{\nu}_{m'-2} & \\ \nu_{m'-2} & \hat{\mu}_{m'-2} & \\ & & I_\delta \end{bmatrix}.$$

- If m is odd, there exists a unitary (2×2) matrix $T_{m'-1}$ such that

  $$T_{m'-1}^H \begin{bmatrix} \pi_{m'-1} \\ \sigma_{m'-1} \nu_{m'-2} \end{bmatrix} = \begin{bmatrix} \sigma_{m'-1}' \end{bmatrix}$$

  with $\pi_{m'-1} = -\hat{\beta}_{m'-2}^H \mu_{m'-2} - \hat{\alpha}_{m'-2}^H \gamma_{m'-1} \nu_{m'-2}$ and $\sigma_{m'-1}'$ upper triangular. Finally we get

  $$\begin{bmatrix} I_2 & \\ & T_{m'-1}^H \end{bmatrix} B \begin{bmatrix} I_2 & \\ & T_{m'-1} \end{bmatrix} = \begin{bmatrix} -\gamma_{m'-1}' & \hat{\sigma}_{m'-1}' \\ \sigma_{m'-1}' & \hat{\sigma}_{m'-1}' \end{bmatrix}.$$

  The last unitary matrix $G_{m'-1}'$ is built up by these block Schur parameters.

- If m is even, there exists a (3×3) unitary matrix $T_{m'-1}$ such that

  $$T_{m'-1}^H \begin{bmatrix} \pi_{m'-1} \\ \sigma_{m'-1} \nu_{m'-2} \end{bmatrix} = \begin{bmatrix} \sigma_{m'-1}' \\ 0 \end{bmatrix}$$

  with $\pi_{m'-1}$ as above and $\sigma_{m'-1}'$ upper triangular. Finally we get

  $$\begin{bmatrix} I_2 & \\ & T_{m'-1}^H \end{bmatrix} B \begin{bmatrix} I_2 & \\ & T_{m'-1} \end{bmatrix} = \begin{bmatrix} -\gamma_{m'-1}' & \hat{\sigma}_{m'-1}' & \\ \sigma_{m'-1}' & \hat{\sigma}_{m'-1}' & \\ & & 1 \end{bmatrix} \begin{bmatrix} I_2 & & \\ & -\gamma_{m'}' & \hat{\sigma}_{m'}' \\ & \sigma_{m'}' & \hat{\gamma}_{m'}' \end{bmatrix}.$$

  These block Schur parameters give us the final unitary matrices $G_{m'-1}'$ and $G_{m'}'$.

4 The solution of the rational approximation problem

After applying m steps of the algorithm constructed above, we get $Q^H Z Q = H$ and $Q^H \begin{bmatrix} f' & -e' \end{bmatrix} = \begin{bmatrix} \sigma_0^T & 0 \end{bmatrix}^T$ with H unitary block Hessenberg given by its block Schur parameters, γ_k, σ_k, $\hat{\sigma}_k$ and $\hat{\gamma}_k$, $k = 1, 2, \ldots, m'-1$. For the moment we also assume that all σ_k are nonsingular, which we shall call the regular case. If some σ_k is singular, we call it the singular case.

4.1 The regular case

When the block Schur parameters σ_k are regular, the block Schur parameters allow us to define the orthonormal block polynomials ϕ_k in a recursive way. First, we need the following properties for the block columns of Q based on the same Schur parameters.

Theorem 4.1 *The block columns of Q satisfy the following recursion*

$$\begin{aligned} Q_{k+1}\sigma_k &= ZQ_k + Q'_k\gamma_k \\ Q'_{k+1}\hat{\sigma}_k^H &= ZQ_k\gamma_k^H + Q'_k, \qquad k = 1, 2, 3, \ldots, m'-1 \end{aligned}$$

with initialization $Q_1 = Q'_1 = [f' \ -e']\,\sigma_0^{-1}$.

Proof. From $Q^H\,[f' \ -e'] = \left[\sigma_0^T \ 0\right]^T$, we get that $Q_1 = [f' \ -e']\,\sigma_0^{-1}$. We set $Q'_1 = Q_1$.

From $Q^H ZQ = H$, we get $ZQ = QH = QG_1G_2\cdots G_{m'-1}$. Denote the $(k+1)$-st block column of $QG_1G_2G_k$ by Q'_{k+1}. Hence, by comparing the block columns of $ZQ = QH$, we get the following recursion

$$\begin{aligned} ZQ_k &= -Q'_k\gamma_k + Q_{k+1}\sigma_k \\ Q'_{k+1} &= Q'_k\hat{\sigma}_k + Q_{k+1}\hat{\gamma}_k. \end{aligned}$$

We get Q_{k+1} out of the first equation and use this result to eliminate Q_{k+1} from the second equation. Hence,

$$\begin{aligned} Q_{k+1} &= (ZQ_k + Q'_k\gamma_k)\sigma_k^{-1} \\ Q'_{k+1} &= ZQ_k\sigma_k^{-1}\hat{\gamma}_k + Q'_k(\hat{\sigma}_k + \gamma_k\sigma_k^{-1}\hat{\gamma}_k). \end{aligned}$$

Because

$$\begin{bmatrix} -\gamma_k & \hat{\sigma}_k \\ \sigma_k & \hat{\gamma}_k \end{bmatrix}\begin{bmatrix} -\gamma_k^H & \sigma_k^H \\ \hat{\sigma}_k^H & \hat{\gamma}_k^H \end{bmatrix} = \begin{bmatrix} I_2 & 0 \\ 0 & I_2 \end{bmatrix},$$

we can rewrite $\sigma_k^{-1}\hat{\gamma}_k$ as $\gamma_k^H\hat{\sigma}_k^{-H}$ and $\hat{\sigma}_k + \gamma_k\sigma_k^{-1}\hat{\gamma}_k$ as $\hat{\sigma}_k^{-H}$. This proves the theorem. □

We define the block polynomials $\phi_k, k = 0, 1, 2, \ldots$ by a similar recursion and show that these block polynomials are orthonormal with respect to the scalar product

$$\langle A(z), B(z)\rangle_v = \sum_{i=1}^{m} \overline{A(z_i)}^T v_i B(z_i).$$

Definition 4.1 *We define the block polynomials ϕ_k and ϕ'_k, $k = 0, 1, 2, \ldots$ by the following recursion based on the block Schur parameters*

$$\begin{aligned} \phi_k(z)\sigma_k &= z\phi_{k-1}(z) + \phi'_{k-1}(z)\gamma_k \\ \phi'_k(z)\hat{\sigma}_k^H &= z\phi_{k-1}(z)\gamma_k^H + \phi'_{k-1}(z), \qquad k = 1, 2, \ldots, m'-1 \end{aligned} \tag{4.1}$$

with the initialization $\phi_0 = \phi'_0 = \sigma_0^{-1}$.

Property 4.2 *The block polynomials ϕ_k and ϕ'_k satisfy the following properties (with $\hat{\sigma}_0 = \sigma_0^H$):*

- *The polynomial ϕ_k has degree k, highest degree coefficient $\sigma_0^{-1}\sigma_1^{-1}\cdots\sigma_{k-1}^{-1}\sigma_k^{-1}$ and constant coefficient $\hat{\sigma}_0^{-H}\hat{\sigma}_1^{-H}\cdots\hat{\sigma}_{k-1}^{-H}\hat{\sigma}_k^{-H}\hat{\gamma}_k^H$.*
- *The polynomial ϕ'_k has degree k, highest degree coefficient $\sigma_0^{-1}\sigma_1^{-1}\cdots\sigma_k^{-1}\hat{\gamma}_k$ and constant coefficient $\hat{\sigma}_0^{-H}\hat{\sigma}_1^{-H}\cdots\hat{\sigma}_{k-1}^{-H}\hat{\sigma}_k^{-H}$.*
- *The block polynomials ϕ_k and ϕ'_k are connected to the block columns Q_k and Q'_k as follows*

$$\begin{aligned} Q_{k+1} &= D\Phi_k \\ Q'_{k+1} &= D\Phi'_k \end{aligned}$$

with $D = \mathrm{diag}\{[f'_1 \ -e'_1], \ldots [f'_m \ -e'_m]\}$ and $\Phi_k = \left[\phi_k(z_1)^T \ \phi_k(z_2)^T \cdots \phi_k(z_m)^T\right]^T$ (Φ'_k is defined similarly).

Proof. The properties immediately follow from the recurrence relations for the block polynomials ϕ_k and ϕ'_k and the fact that $\gamma_k, \sigma_k, \hat{\sigma}_k, \hat{\gamma}_k$ are block Schur parameters. □

The following result allows us to solve the discrete linearized least squares rational approximation problem on the unit circle.

Theorem 4.3 (orthonormality) *The block polynomials ϕ_k as defined above are orthonormal with respect to the scalar product $\langle \cdot, \cdot \rangle_v$, i.e. $\langle \phi_i(z), \phi_j(z) \rangle_v = \delta_{i,j} I_2$.*

Proof. Using the results of the previous theorem, the scalar product can also be written as

$$\begin{aligned} \langle \phi_i(z), \phi_j(z) \rangle_v &= \Phi_i D^H D \Phi_j \\ &= Q_{i+1}^H Q_{j+1} \\ &= \delta_{i,j} I_2 \end{aligned}$$

because $Q^H Q = I_m$. □

Let us now return to the rational approximation problem. To solve this problem, we shall write the solution in terms of the orthonormal block polynomials ϕ_j. Because the highest degree coefficients of these block polynomials are upper triangular, we can always write our solution $n(z)/d(z)$ of degree α as follows

$$\begin{bmatrix} d(z) \\ n(z) \end{bmatrix} = \sum_{j=0}^{\alpha} \phi_j(z) \begin{bmatrix} \rho_j \\ \tau_j \end{bmatrix},$$

with $(\rho_\alpha, \tau_\alpha) \neq (0,0)$. Remember that the solution we are looking for, has to be normalized. From property 4.2, we know that the highest degree coefficient of ϕ_α is equal to

$$\sigma_0^{-1}\sigma_1^{-1}\cdots\sigma_\alpha^{-1} = \begin{bmatrix} 1/\prod_{j=0}^{\alpha} s_{1,j} & \times \\ 0 & 1/\prod_{j=0}^{\alpha} s_{2,j} \end{bmatrix}, \quad \text{if } \sigma_j = \begin{bmatrix} s_{1,j} & s_{3,j} \\ 0 & s_{2,j} \end{bmatrix}. \tag{4.2}$$

The specific value of $\times$ is not important for the moment. To normalize $n(z)/d(z)$, we have to choose $\tau_\alpha = \prod_{j=0}^{\alpha} s_{2,j}$ if $n_\alpha \neq 0$ or $\rho_\alpha = \prod_{j=0}^{\alpha} s_{1,j}$ if $n_\alpha = 0$. To solve the rational approximation problem, we have to minimize

$$\sum_{i=1}^{m} |f'_i d(z_i) - e'_i n(z_i)|^2 = \sum_{k=0}^{\alpha} \left(|\rho_k|^2 + |\tau_k|^2 \right).$$

As we have seen in Section 2, we can make this entity smaller by putting ρ_k and τ_k equal to zero for $k = 0, 1, \ldots, \alpha - 1$. This reduces the distance between $(n(z), d(z))$ and the data to $\sqrt{|\rho_\alpha|^2 + |\tau_\alpha|^2}$. The way in which we have stated the rational approximation problem requires $n(z)/d(z)$ to have degree α. From the normalization conditions, it is clear that we should therefore take the "right" structure, i.e.

$$\tau_\alpha = \prod_{j=0}^{\alpha} s_{2,j} \text{ and } \rho_\alpha = 0, \quad \text{if } \prod_{j=0}^{\alpha} |s_{2,j}| \leq \prod_{j=0}^{\alpha} |s_{1,j}|$$

and the "left" structure, i.e.

$$\rho_\alpha = \prod_{j=0}^{\alpha} s_{1,j} \text{ and } \tau_\alpha = 0, \quad \text{if } \prod_{j=0}^{\alpha} |s_{2,j}| \geq \prod_{j=0}^{\alpha} |s_{1,j}|$$

where, in case of equality of the two distances, we can choose between the two solutions. Note that this choice corresponds to our problem setting, but that we have the possibility to solve also slightly different problems in an equally simple way. E.g., we could have required $n(z)/d(z)$ to be strictly proper, i.e. with $n_\alpha = 0$ and $d_\alpha = 1$. Then we had to choose the second solution, i.e. the one with $\tau_\alpha = 0$. It is easy to generalize this to more complicated structures than the strictly proper one. Suppose for example that we look for a solution with $n_\alpha = 0$, $n_{\alpha-1} = 0$ and $d_\alpha = 1$. From $n_\alpha = 0$, we easily derive that $\tau_\alpha = 0$. From $d_\alpha = 1$, it follows that $\rho_\alpha = \prod_{j=0}^{\alpha} s_{1,j}$. To make $n_{\alpha-1}$ equal to zero, we have to take $\tau_{\alpha-1}$ equal to a specific value, possibly different from zero. We shall not follow this path in the sequel but concentrate our attention on the original problem setting. The other cases are quite similar. Let us investigate how good the rational approximation is.

Theorem 4.4 *If we write our rational approximant $n(z)/d(z)$ in terms of the parameters $\rho_k, \tau_k, k = 0, 1, 2, \ldots, \alpha$, then the approximation error made in the ith point $z_i, i = 1, 2, \ldots, m$ can be measured as follows*

$$f_i' d(z_i) - e_i' n(z_i) = \sum_{k=0}^{\alpha} Q_{i,k+1} \begin{bmatrix} \rho_k \\ \tau_k \end{bmatrix}.$$

Proof. This result follows immediately from the fact that

$$\begin{bmatrix} f_i' & -e_i' \end{bmatrix} \phi_k(z_i) = Q_{i,k+1},$$

the i-th row of the block column Q_{k+1}. □

For example, if we take the strictly proper solution, all ρ_k and τ_k values are zero except one, $\rho_\alpha = \prod_{j=0}^{\alpha} s_{1,j}$. The approximation error in each point z_i can thus be written as

$$f_i' d(z_i) - e_i' n(z_i) = Q_{i,\alpha+1} \begin{bmatrix} \rho_\alpha \\ 0 \end{bmatrix},$$

i.e. the error is given by a scaled version of the first column of $Q_{\alpha+1}$.

4.2 The singular case

Until now, we have taken for granted that all the subdiagonal blocks σ_j were nonsingular. What will happen if one of the upper triangular matrices σ_j is singular, i.e. if $s_{1,j}$ or $s_{2,j}$ is equal to zero?

Theorem 4.5 *We shall assume that σ_α is the first block Schur parameter with $s_{1,\alpha}$ or (if it exists) $s_{2,\alpha}$ equal to zero. We define the block polynomial $\tilde{\phi}_\alpha$ as follows*

$$\tilde{\phi}_\alpha = z\phi_{\alpha-1}(z) + \phi_{\alpha-1}'(z)\gamma_\alpha.$$

If $s_{1,\alpha} = 0$, then

$$\begin{bmatrix} d(z) \\ n(z) \end{bmatrix} = \tilde{\phi}_\alpha(z) \begin{bmatrix} \prod_{j=0}^{\alpha-1} s_{1,j} \\ 0 \end{bmatrix}$$

gives us a normalized rational interpolant $n(z)/d(z)$ with $d_\alpha = 1$.

Otherwise, $s_{2,\alpha} = 0$ and

$$\begin{bmatrix} d(z) \\ n(z) \end{bmatrix} = \tilde{\phi}_\alpha(z) \begin{bmatrix} -s_{3,\alpha}/s_{1,\alpha} \\ 1 \end{bmatrix} \prod_{j=0}^{\alpha-1} s_{2,\alpha}$$

gives us a normalized rational interpolant $n(z)/d(z)$ with $n_\alpha = 1$.

Proof. Because the previous block Schur parameters $\sigma_j, j = 0, 1, 2, \ldots \alpha - 1$ are regular, we can compute $\tilde{\phi}_\alpha$ based on $\phi_{\alpha-1}$ and $\phi_{\alpha-1}'$. Moreover, $\tilde{\phi}_\alpha$ has degree α with the same highest degree coefficient as $\phi_{\alpha-1}$, i.e. $\prod_{j=0}^{\alpha-1} \sigma_j^{-1}$.

Using the relationship between the block columns of the unitary matrix Q and the orthonormal block polynomials $\phi_j, j = 0, 1, \ldots, \alpha - 1$, we get

$$Q_{i,\alpha+1}\sigma_\alpha = [f_i' \ -e_i'] \tilde{\phi}_\alpha(z_i), \qquad i = 1, 2, \ldots, m$$

with

$$\sigma_\alpha = \begin{bmatrix} s_{1,\alpha} & s_{3,\alpha} \\ 0 & s_{2,\alpha} \end{bmatrix}.$$

If we summarize the two cases as

$$\begin{bmatrix} d(z) \\ n(z) \end{bmatrix} = \tilde{\phi}_\alpha(z) \begin{bmatrix} \tilde{\rho} \\ \tilde{\tau} \end{bmatrix},$$

we get that

$$\begin{aligned} [f_i' \ -e_i'] \begin{bmatrix} d(z_i) \\ n(z_i) \end{bmatrix} &= [f_i' \ -e_i'] \tilde{\phi}_\alpha(z_i) \begin{bmatrix} \tilde{\rho} \\ \tilde{\tau} \end{bmatrix} \\ &= Q_{i,\alpha+1}\sigma_\alpha \begin{bmatrix} \tilde{\rho} \\ \tilde{\tau} \end{bmatrix} \\ &= 0, \qquad i = 1, 2, \ldots, m \end{aligned}$$

for the specific choices of $\tilde{\rho}$ and $\tilde{\tau}$. Hence, $n(z)/d(z)$ is a rational interpolant for the given data. By looking at the highest degree coefficient, we see that this rational interpolant is normalized. This proves the theorem. □

The rational function $n(z)/d(z)$ from the previous theorem approximates the given data exactly, i.e. $n(z)/d(z)$ is an interpolant of the given data. It is clear that this situation will appear rarely in practice when the data are only given with a certain finite precision.

5 Deriving a rational approximant with real coefficients

In a lot of practical applications, e.g. digital filter design, model reduction, modal analysis, system identification, one looks for a rational approximant $n(z)/d(z)$ with real coefficients. The algorithm, as described before, results in complex coefficients in general. If we want $n(z)/d(z)$ to have real coefficients, the data should be consistent with this restriction, i.e. if z is a given abscissa with corresponding estimated function value f/e, then $\overline{f}/\overline{e}$ can be considered as an estimated function value in $\overline{z}$. Using this fact, the initial data matrix M can be transformed using a simple unitary matrix into a similar *real* matrix. From this point on, all computations can be done using real arithmetic, resulting in real block Schur parameters, real block orthogonal polynomials and finally a real rational approximant. The simple unitary matrix to do this initial similarity transformation can be chosen for each data pair in several ways, e.g.

$$Q = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix}$$

because

$$Q^H \begin{bmatrix} f' & -e' \\ \overline{f}' & -\overline{e}' \end{bmatrix} = \begin{bmatrix} \sqrt{2}\,\mathrm{real}(f') & -\sqrt{2}\,\mathrm{real}(e') \\ -\sqrt{2}\,\mathrm{imag}(f') & \sqrt{2}\,\mathrm{imag}(e') \end{bmatrix}$$

and

$$Q^H \begin{bmatrix} e^{j\omega} & 0 \\ 0 & e^{-j\omega} \end{bmatrix} Q = \begin{bmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{bmatrix}.$$

6 Parallel implementation

Regarding the amount of computational work, we can make a distinction between the computation of the unitary block Hessenberg matrix $Q^H Z Q$ from the data and the computation of the orthonormal block polynomials. Fixing the degree $\alpha \ll m$ of the rational approximant implies that we can obtain the rational approximant from $\phi_\alpha(z)$. Hence, it is not necessary to compute the complete block Schur parametrization but only the first part consisting of $G_1, G_2, \ldots, G_\alpha$. Compared to the complexity m^2 of computing the complete block block Schur parametrization, this decreases the amount of computational work to $m\alpha$. The second part of the solution method requires the computation of $\phi_0, \phi_1, \ldots, \phi_\alpha$, which requires a number of floating point operations of the order of α^2. Therefore, since $\alpha \ll m$, it is important to decrease the execution time of the first part by parallel implementation. This could be done by a sort of pipeline method.

Let us assume that we have $\alpha \ll m$ processors. Each processor is coupled to a G_k, $k = 1, 2, \ldots, \alpha$, matrix and updates the corresponding block Schur parameters in each step. Once the kth processor has computed α_k, β_k, $\hat{\alpha}_k$, $\hat{\beta}_k$ and T_k, the $(k+1)$th processor can use this information to update the corresponding block Schur parameters. Once the pipeline is started up, the execution time is equivalent to $O(m)$ floating point operations.

7 Examples

In this section, we give three examples. The first one is rather theoretical, the others are more practical. In the three examples, we look for a rational approximant with *real* coefficients (see Section 5). To measure the goodness of the rational approximant, we consider a relative and absolute approximation error in each point z_i. We take all the weights $w_i^{(p)} = 1$. The absolute error that is plotted in the figures is then nothing else but $|r_i^{(p)}|$. Similarly, when the relative error is plotted, we plot $|r_i^{(p)}/(f_i/e_i)|$. The data is normalized in such a way that $\sum_{i=1}^{m}(|f_i'|^2 + |e_i'|^2) = \|\sigma_0\|_F^2 = 1$ ($\|\cdot\|_F$ stands for the Frobenius-norm). We can do this by computing w_i such that

$$w_i(|f_i|^2 + |e_i|^2) = 1/m \tag{7.3}$$

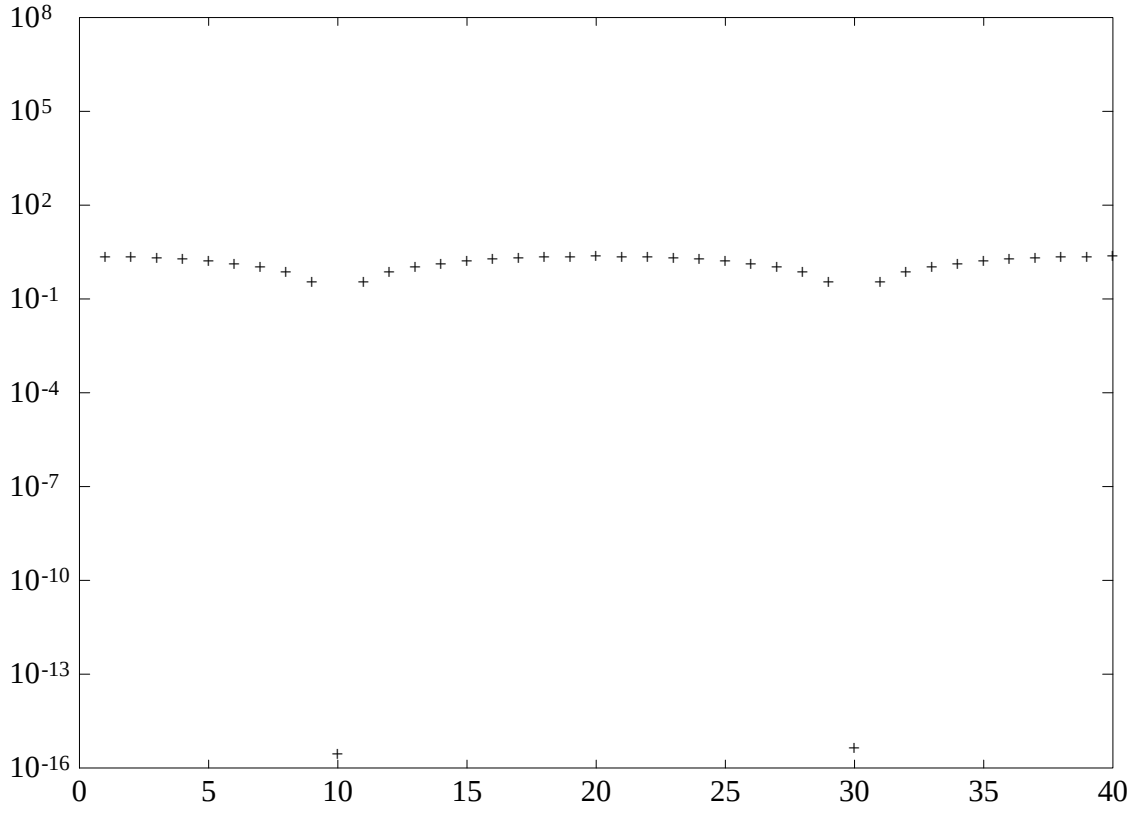


Figure 1: The modulus of the function values of example 1

The value of the rational approximant $n(z_i)/d(z_i)$ is computed using the Horner scheme. Of course, we could also use the recurrence (4.1) where z is replaced by z_i. For equally distributed points z_i on the unit circle, as in the examples here, one could also use FFT. All computations were done in double precision on a DEC workstation using PRO-MATLAB with machine epsilon $eps = 2.2204e-16$.

Example 7.1 In the first example, we consider the $m = 40$ interpolation points $z_i = exp(ij2\pi/m)$, $i = 1, 2, \ldots, m$, with function values

$$f_i/e_i = z_i^3 + z_i$$

i.e., the polynomial $z(z-j)(z+j)$ evaluated at the points z_i. The modulus of these function values is plotted in Figure 1. The errors connected to the "left" and "right" solution of the linearized problem are given in the following table.

j	$\prod_{k=0}^{j} \lvert s_{1,k} \rvert$	$\prod_{k=0}^{j} \lvert s_{2,k} \rvert$
0	$7.4349607e-01$	$6.6874031e-01$
1	$7.2654253e-01$	$5.5589297e-01$
2	$6.1803399e-01$	$5.2573111e-01$
3	$6.1803399e-01$	$6.3059264e-16$
4	$6.0836632e-01$	$5.4738287e-16$
5	$5.8550486e-01$	$5.1239811e-16$

From this table, it is clear that we should choose α equal to 3 and take the "right" structure for the rational approximant. The next table shows the coefficients of the numerator and denominator polynomial of the rational approximant of degree $\alpha = 3$.

degree	numerator coefficients	denominator coefficients
0	$3.152963175230421e-16$	$1.000000000000000e+00$
1	$1.000000000000001e+00$	$1.633234097724558e-16$
2	$8.088600263652522e-17$	$1.576481587615210e-16$
3	$1.000000000000000e+00$	$1.435322868745093e-16$

The absolute and relative errors are indicated in Figure 2. We can say that the rational approximant, we obtained here, is very good, which, for this example of course will not come as a surprise.

In the next example we try to find a good approximant for the exponential function on the unit circle.

Example 7.2 The points z_i are the same as in the first example. The function values are $f_i/e_i = exp(z_i)$, $i = 1, 2, \ldots, m$. When looking at the errors of the solutions of the linearized problem, we choose to take a "right" solution of degree 3. The absolute and relative errors are given in Figure 3. By executing one iteration step of the Loeb-Wittmeyer algorithm adapting the weights w_i as described in Section 2, we get the errors of Figure 4. Note that, using the normalization of (7.3), the relative error is almost a constant over the set of points z_i. However, doing one iteration step, it is the absolute error having this property. A similar phenomenon can also be observed in the next example. Moreover, the winding number of the absolute error function $exp(z) - n(z)/d(z)$ on the unit circle is equal to $(\deg n + \deg d + 1)$. In several examples, the error function for a best Chebyshev rational approximant n'/d' closely approximates a perfect circle about the origin having a winding number $(\deg n' + \deg d' + 1)$ [13]. Hence, the rational approximant obtained here after doing one iteration step, will be a close candidate to a best Chebyshev rational approximant on the unit circle.

A more practical example is the following one.

Example 7.3 As in the previous examples, we assume the points z_i are equally distributed on the unit circle, i.e. we take $z_i = exp(ij2\pi/m)$, $i = 1, 2, \ldots, m$ with $m = 200$. We choose in a certain random way 5 zeros with modulus between 0 and 0.99. Similarly, we choose 3 poles with modulus between 0.90 and 0.99 and 2 poles with modulus between 0 and 0.90. With these poles and zeros, a rational function is constructed with real coefficients. Hence, we add the complex conjugates of the poles and the zeros. The data f_i/e_i is equal to this rational function evaluated in z_i but we have added a uniformly distributed absolute error of 10^{-4}. The modulus of the function values is shown in Figure 5. Examining the errors connected to the linearized problem, we choose for a "right" solution of a reduced degree 8. The absolute and relative errors are given in Figure 6. Doing one iteration as described in Section 2, we obtain the absolute and relative errors given in Figure 7. The left side of Figure 8 shows the poles (indicated by $\star$) and the zeros (indicated by $\circ$) of the original rational function of degree 10 in the complex plane. The right side shows the poles and zeros of the rational approximant after doing one iteration step. Note that the dominant poles and zeros, i.e. the ones closely to the unit circle, are well approximated.

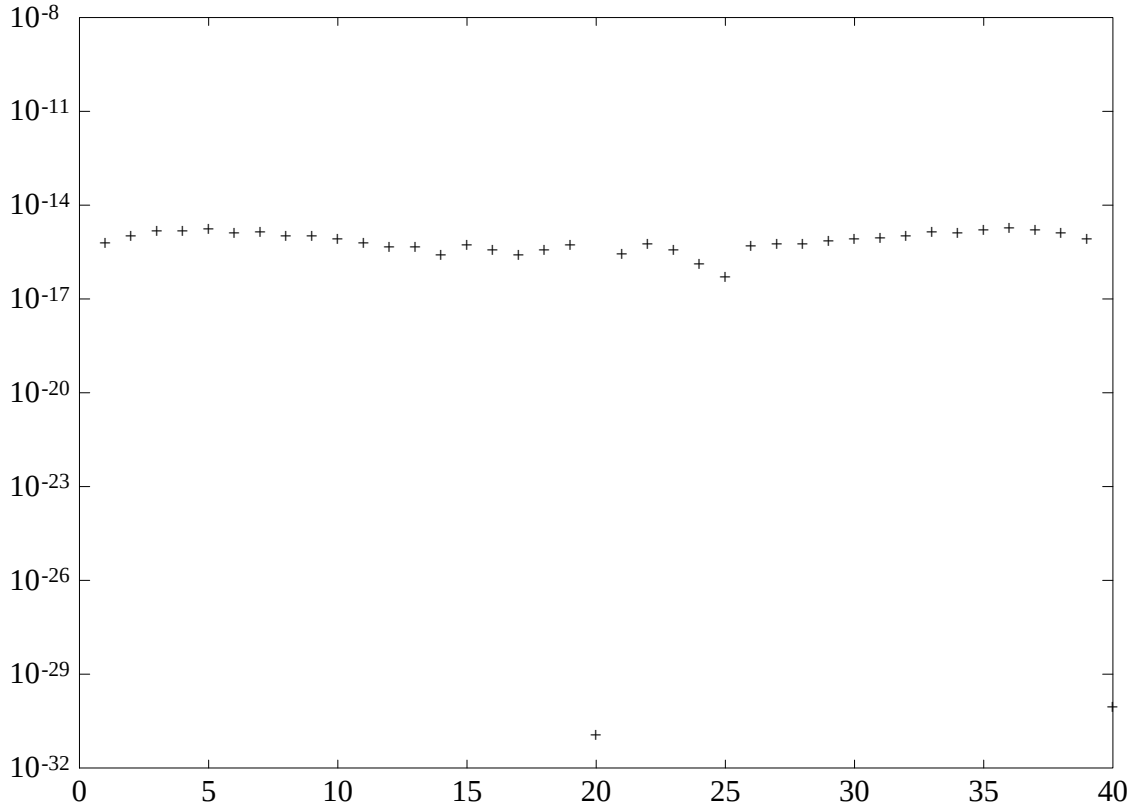
These three examples show that the rational approximant approximates the given data quite well in the proper rational sense. Moreover, applying only one iteration step of the algorithm of Loeb-Wittmeyer already gives an absolute error whose magnitude does not change very much anymore over the set of points z_i.

Because the amount of computational work is small ($O(m\alpha)$ and, in parallel, $O(m)$ floating point operations), the designer of a rational approximant has a lot of freedom in choosing the degree, the iteration method (how are the weights adapted), the number of iteration steps, ... The errors connected to the "left" and "right" solutions of the linearized problem give a good indication for the designer to choose between the several possible structures for the rational approximant (degree, "left" or "right").

In the context of discrete-time linear systems, we can interpret the rational approximation problem solved in this paper as a system identification problem. The data f_i/e_i can be considered as measurements of the frequency response of a discrete-time system. We are now looking for a linear model (a rational function) of limited complexity (small degree) such that the measured frequency response and the frequency response of the model almost coincide (taking into account the error on the measurements). For an introduction to system identification one could look at e.g. [11, chapter 6] and [7, chapter 8]. The rational approximation problem can also be seen as a model reduction problem. In the last example, we start with a model of degree 10 and we approximate this by a model of degree 8 ([5, section 19]).

8 Conclusion

We have developed an efficient method to solve the discrete linearized least squares rational approximation problem. We have shown that this method can be implemented in a pipelined way on a parallel architecture.



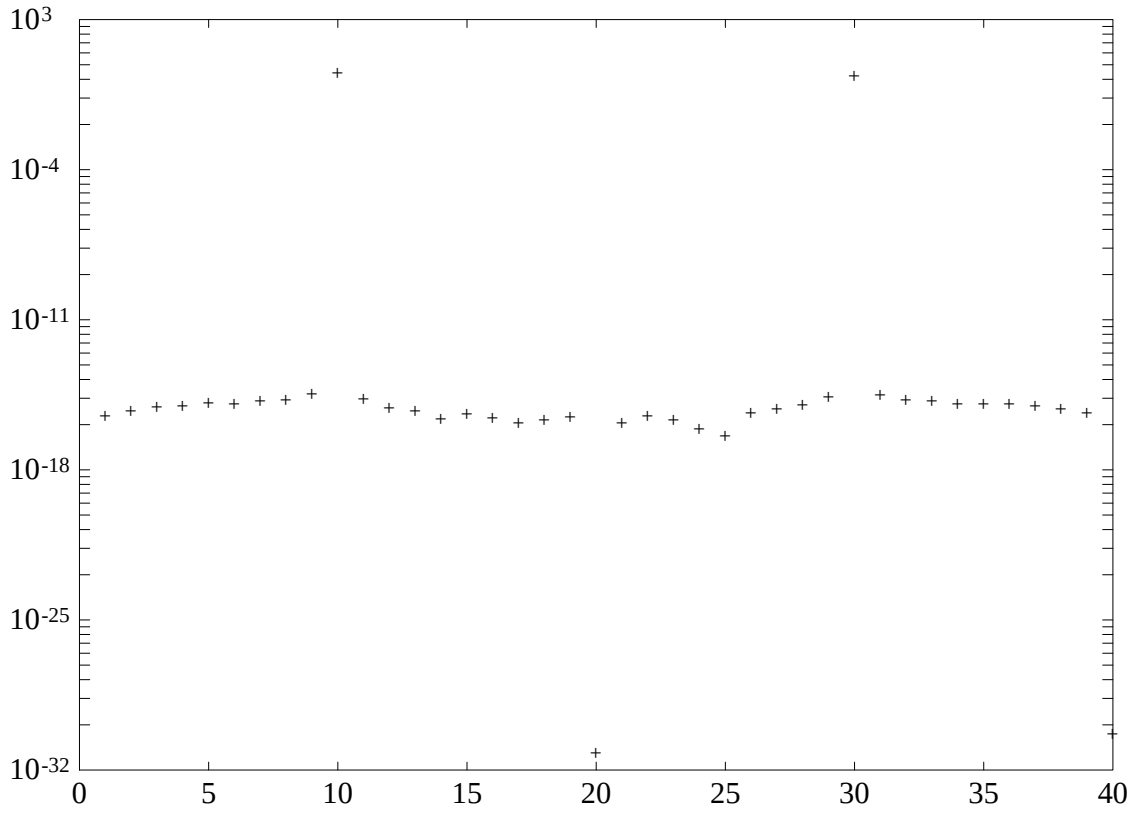
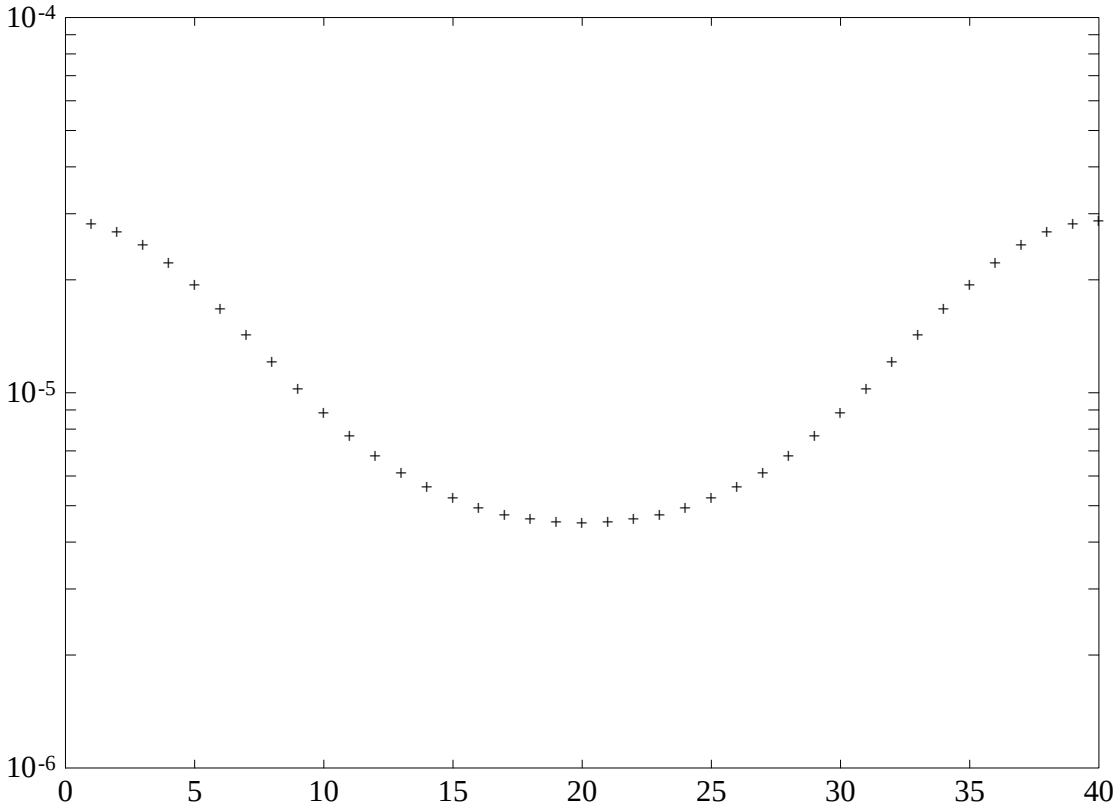


Figure 2: The absolute and relative errors of example 1



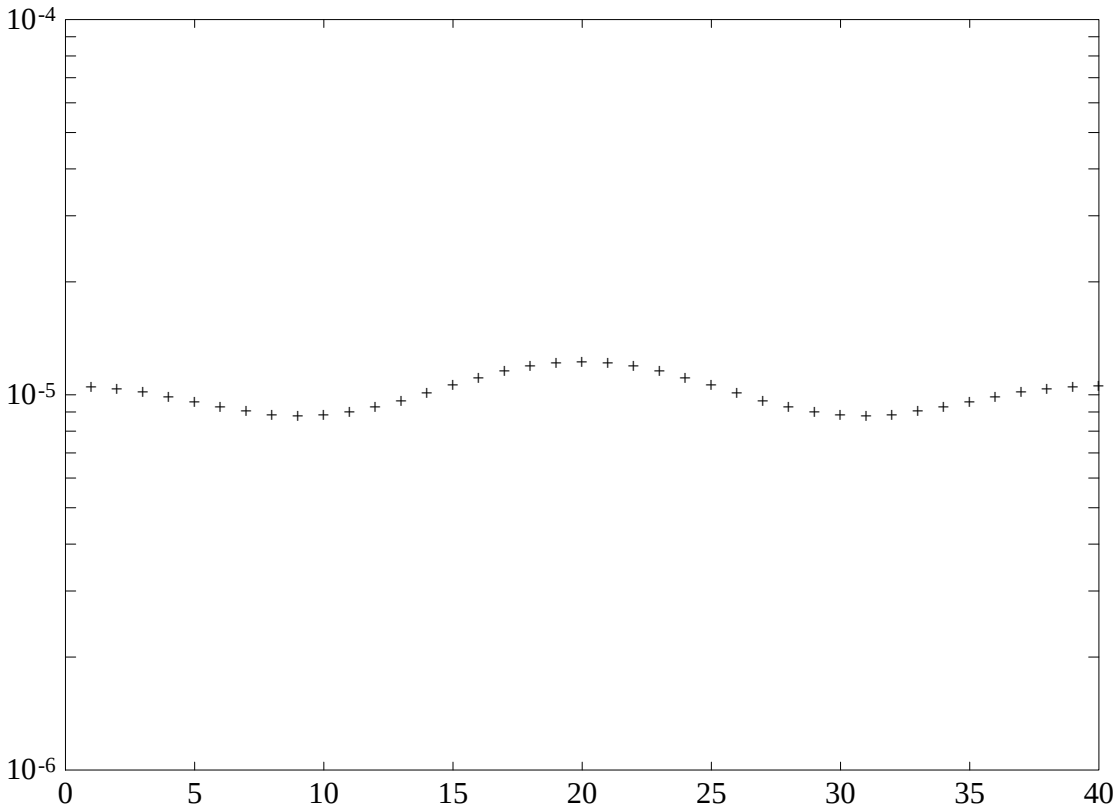
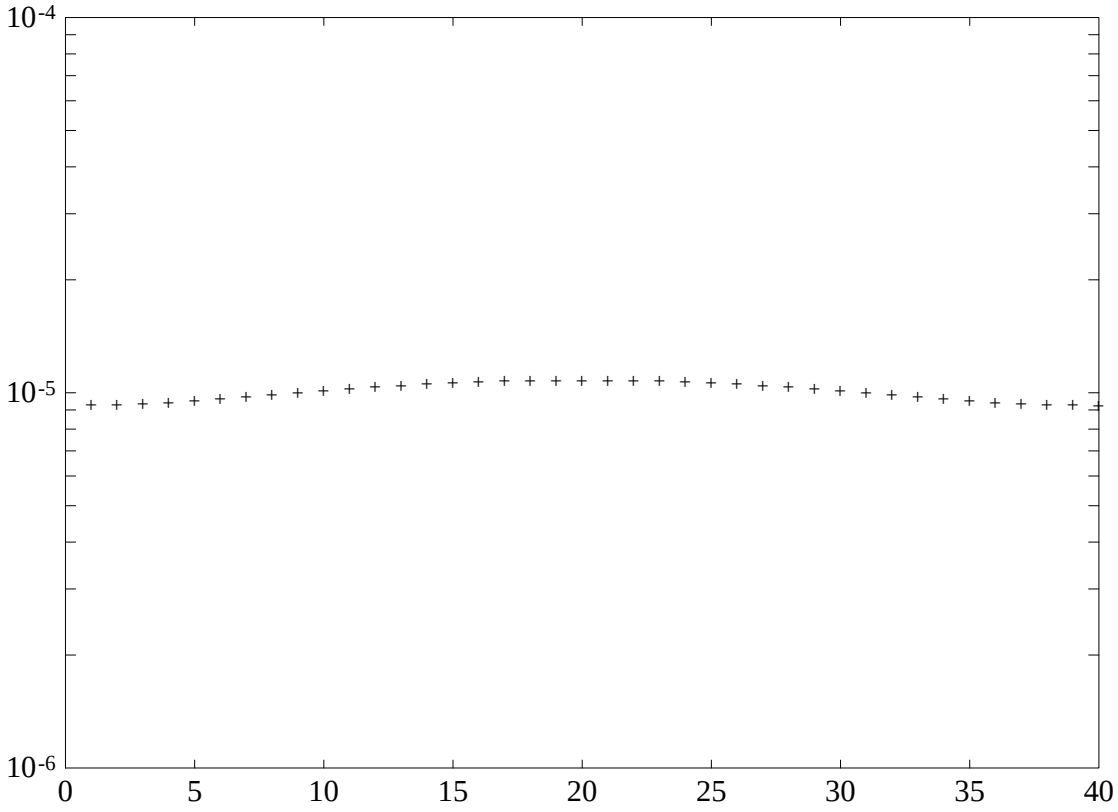


Figure 3: The absolute and relative errors of example 2



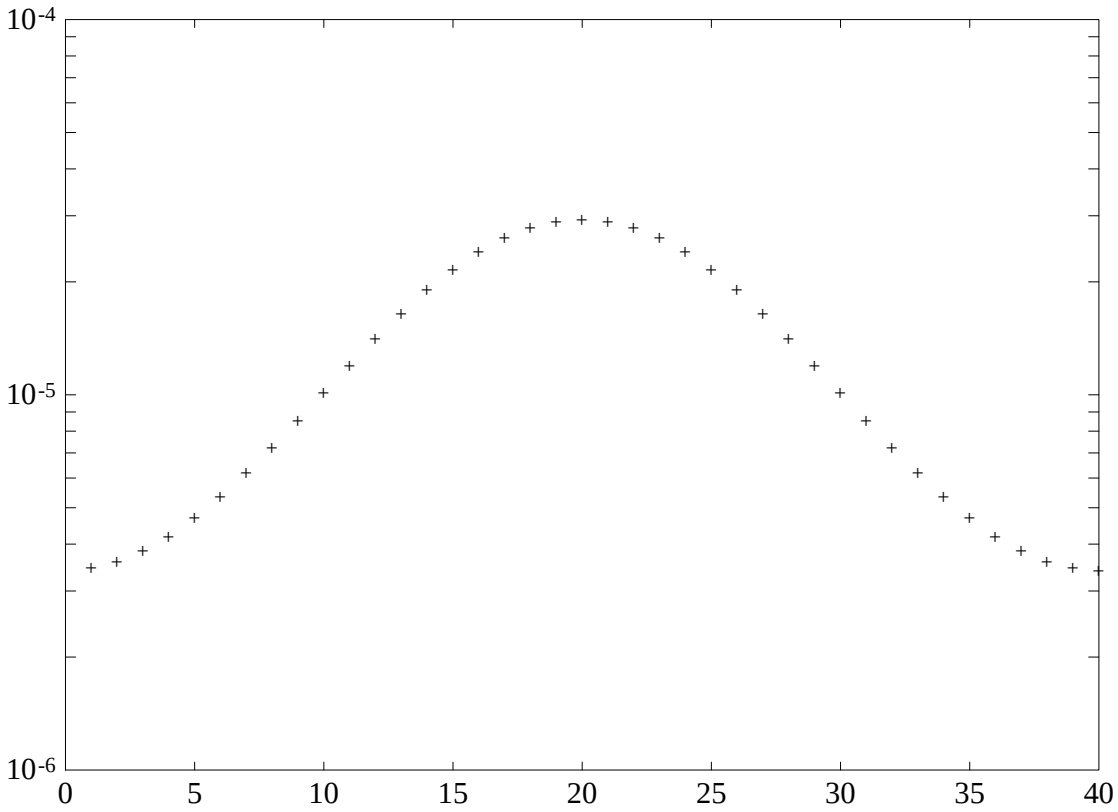


Figure 4: The absolute and relative errors of example 2

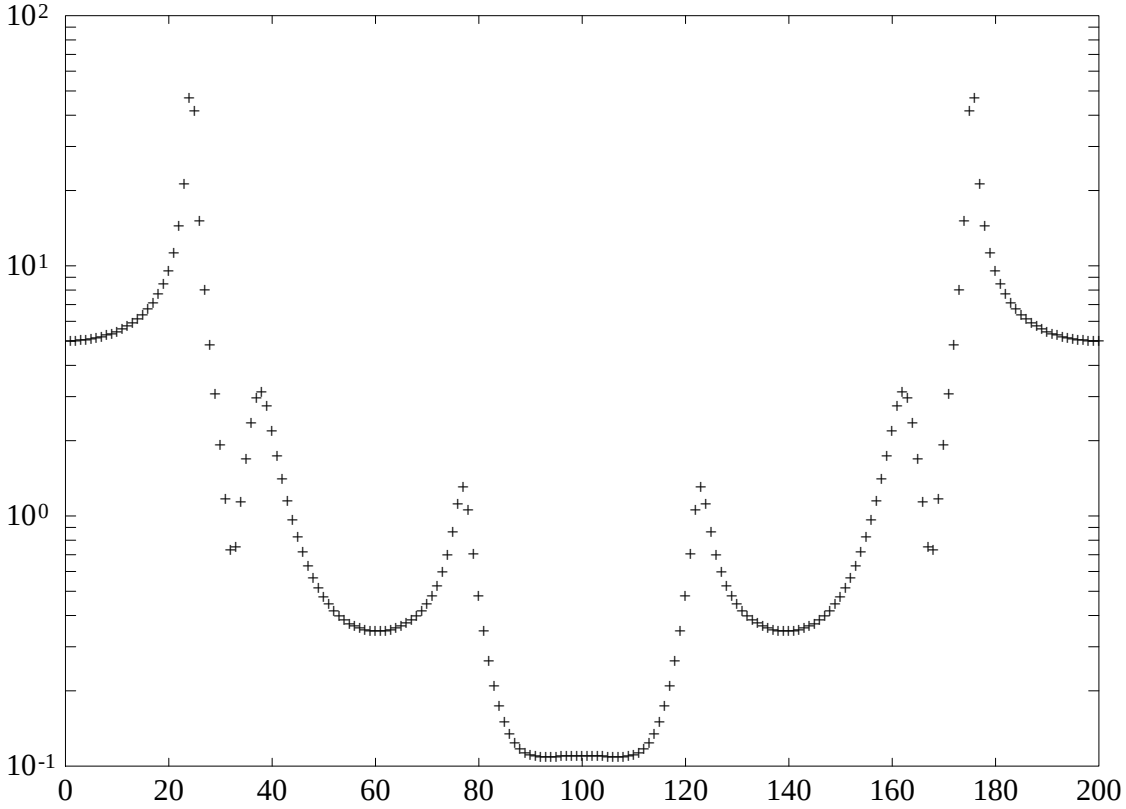
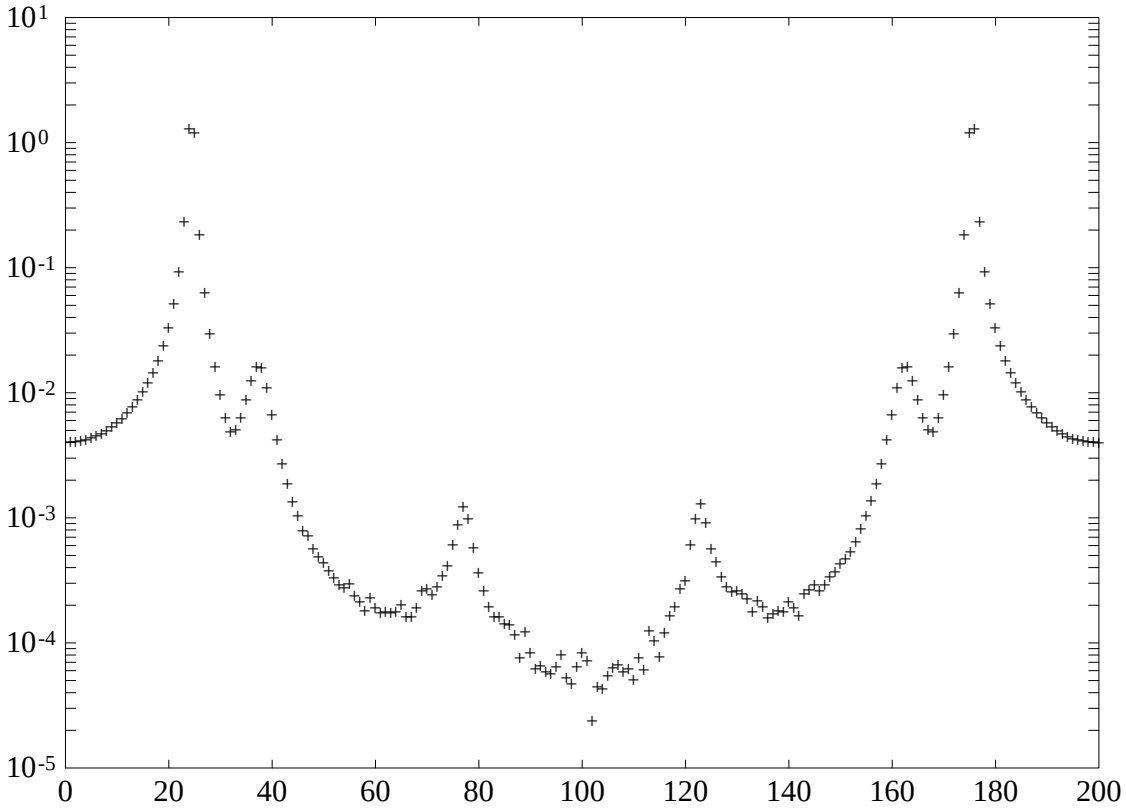


Figure 5: The modulus of the function values of example 3



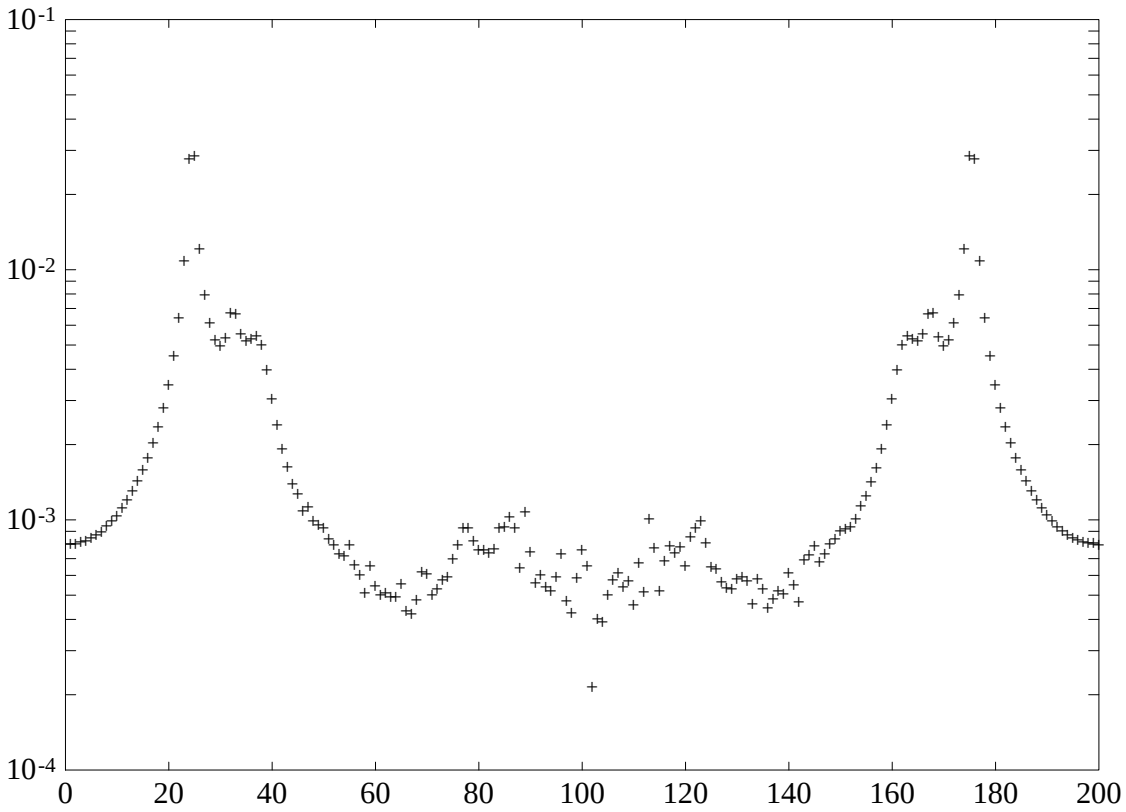
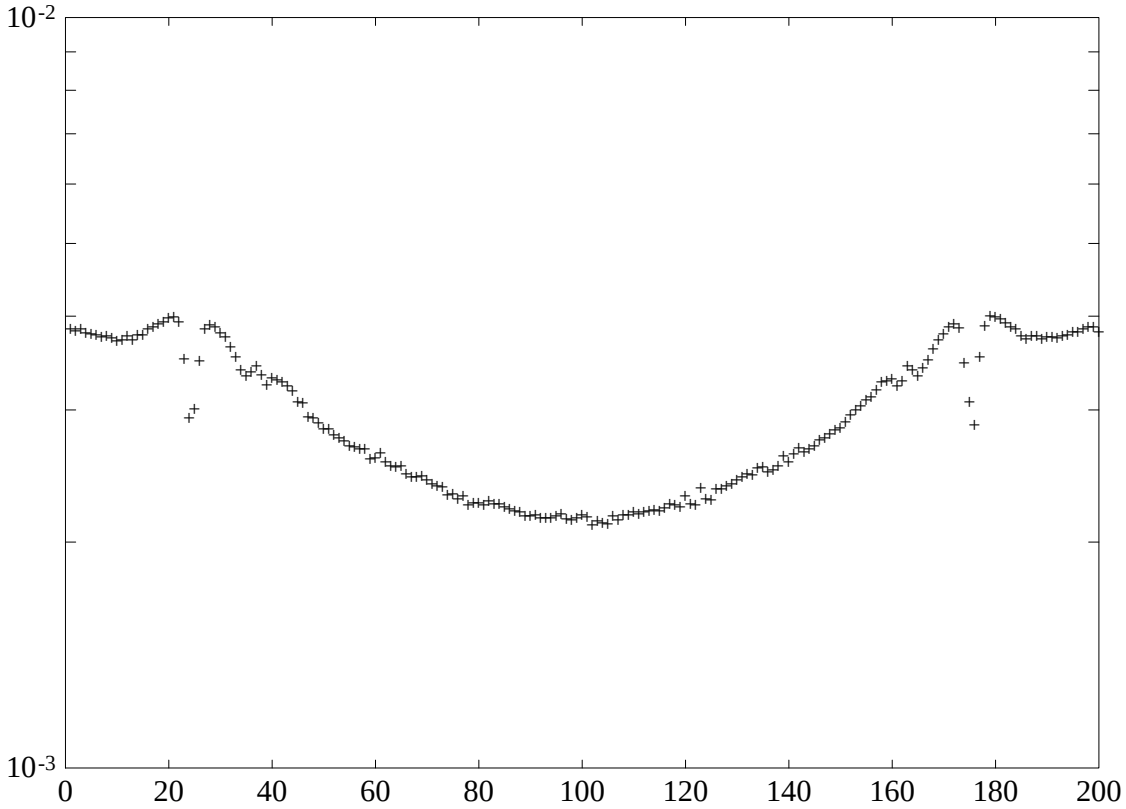


Figure 6: The absolute and relative errors of example 3



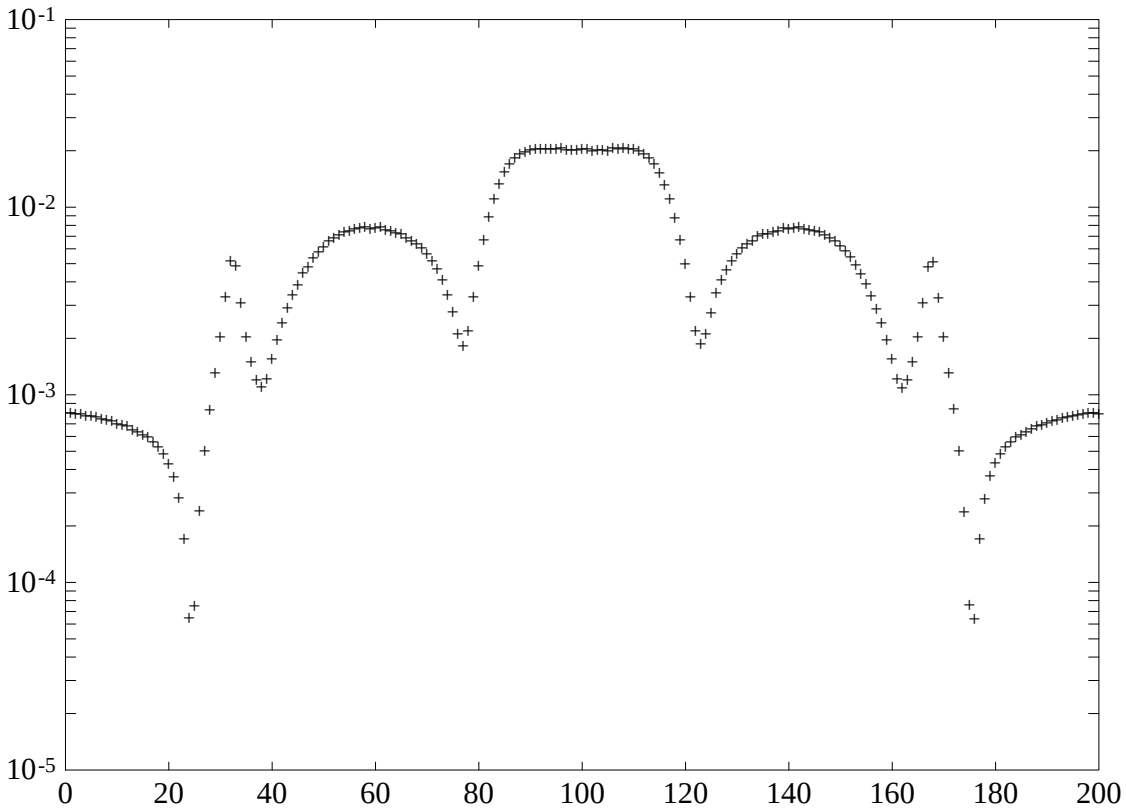
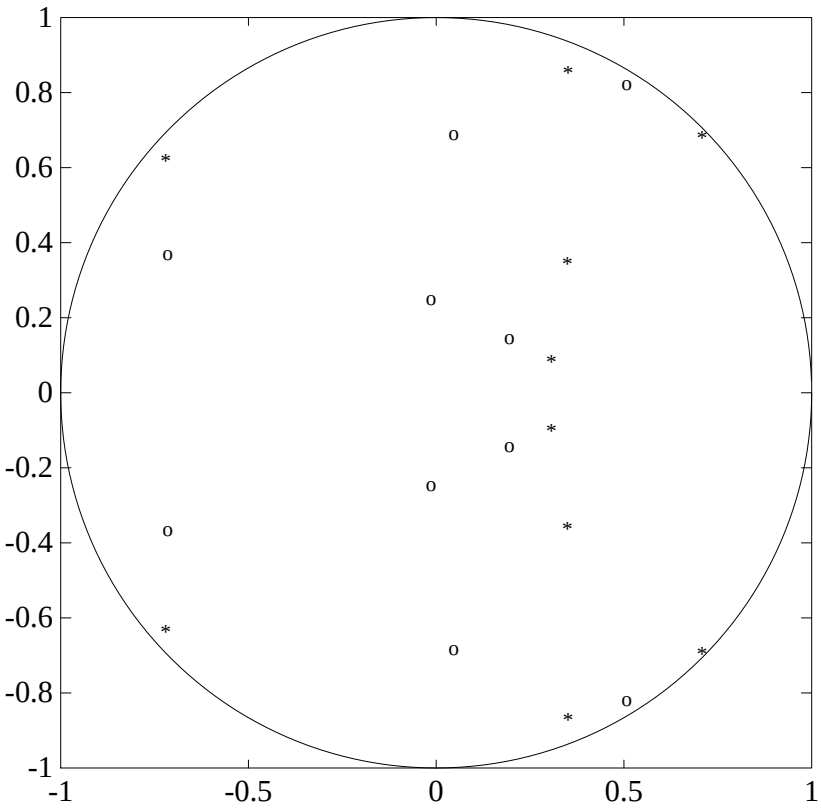


Figure 7: The absolute and relative errors of example 3



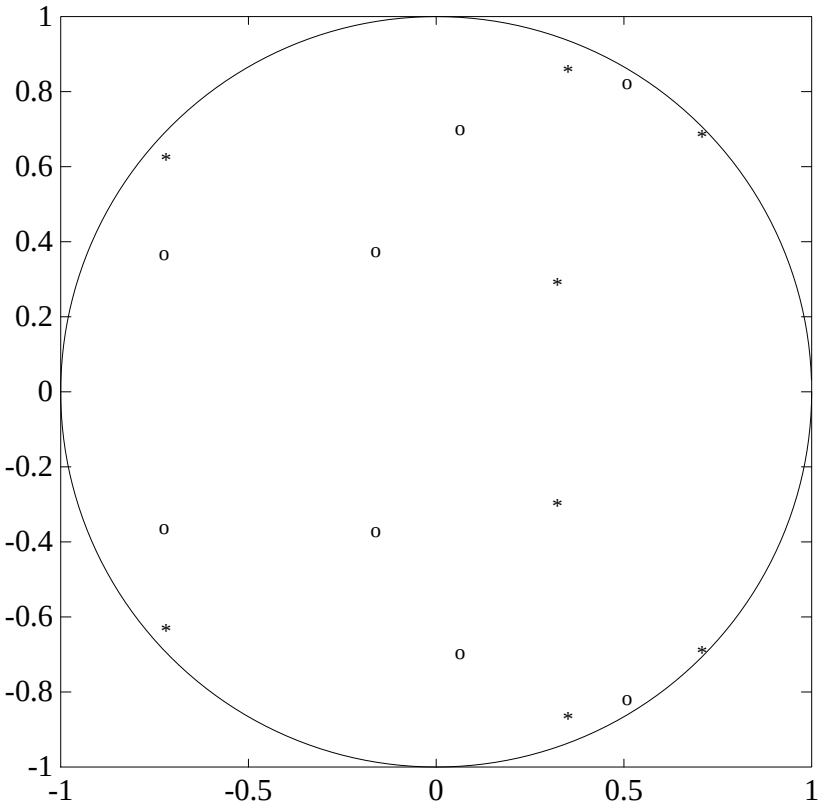


Figure 8: The poles and zeros of the original and the approximating rational function of example 3

The numerical examples show that the computed rational approximant is also quite good as a solution of the proper rational approximation problem. It becomes a very good proper rational approximant after only one iteration step of the algorithm of Loeb-Wittmeyer.

Because the algorithm developed here is so efficient, it could be used as the kernel of a graphical interactive program to design rational approximants for large data sets. As domains of application, we mention digital filter design, modal analysis, model reduction, system identification,....

In future work, this algorithm will be generalized to other structures for the rational approximant, i.e. the maximum degree of numerator and denominator polynomial can be chosen freely. Also more than two data columns f' and e' will be allowed, leading to block orthonormal polynomials with the dimension of the blocks bigger than 2.

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.H. Golub and P. Van Dooren, editors, *Proceedings of the NATO Advanced Study Institute on Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms, Leuven, Belgium, 1988*, pages 385–395. Springer-Verlag, 1991.

[3] I. Barrodale and J.C. Mason. Two simple algorithms for discrete rational approximation. *Mathematics of Computation*, 24(112):877–891, 1970.

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

[5] A. Bultheel and M. Van Barel. Padé techniques for model reduction in linear system theory: a survey. *J. Comp. Appl. Math.*, 14:401–438, 1986.

[6] 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.

[7] P. Faurre and M. Depeyrot. *Elements of system theory.* North-Holland, Amsterdam, 1977.

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

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

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

[11] W.J. Rugh. *Mathematical description of linear systems.* Marcel Dekker, New York, 1975.

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

[13] L.N. Trefethen. Near-circularity of the error curve in complex chebyshev approximation. *J. Approx. Theory*, 31:344–367, 1981.

[14] M. Van Barel and A. Bultheel. A new approach to the rational interpolation problem. *J. Comp. Appl. Math.*, 32:281–289, 1990.

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

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