A unified superfast algorithm for boundary rational tangential interpolation problems and for inversion and factorization of dense structured matrices *

Vadim Olshevsky
Department of Mathematics and CS
Georgia State University
Atlanta, GA 30303
volshevsky@cs.gsu.edu
http://www.cs.gsu.edu/~matvro

Victor Pan
Department of Mathematics and CS
Lehman College, CUNY
Bronx, NY
VPAN@lcvax.lehman.cuny.edu

Abstract

The classical scalar Nevanlinna-Pick interpolation problem has a long and distinguished history, appearing in a variety of applications in mathematics and electrical engineering. There is a vast literature on this problem and on its various far reaching generalizations; for a quick historical survey see [1] and [38]. It is widely known that the now classical algorithm for solving this problem proposed by Nevanlinna in 1929 can be seen as a way of computing the Cholesky factorization for the corresponding Pick matrix. Moreover, the classical Nevanlinna algorithm takes advantage of the special structure *of the Pick matrix to compute this triangular factorization in only $O(n^2)$ arithmetic operations, where n is the number of interpolation points, or, equivalently, the size of the Pick matrix. Since the* structure-ignoring *standard Cholesky algorithm [though applicable to the wider class of general matrices] has much higher complexity $O(n^3)$, the Nevanlinna algorithm is an example of what is now called* fast algorithms. *In this paper we use a divide-and-conquer approach to propose a new* superfast $O(n \log^3 n)$ *algorithm to construct solutions for the more general* boundary tangential *Nevanlinna-Pick problem. This dramatic speed-up is achieved via a new divide-and-conquer algorithm for factorization of rational matrix functions; this superfast algorithm seems to have a practical and theoretical significance itself. It can be used to solve similar rational interpolation problems [e.g., the matrix Nehari problem], and a variety of engineering problems. It can also be used for inversion and triangular factorization of matrices with* displacement structure, *including Hankel-like, Vandermonde-like, and Cauchy-like matrices.*

*This work was supported by NSF grants CCR 9732355, CCR 9625344, CCR 9732206, GSU ORSP award, and PSC CUNY award 669363.

1. Introduction

1.1. Motivation: rational vs. polynomial interpolation. Analysis of the arithmetic complexity of various interpolation problems is a classical topic, and several efficient superfast methods developed by a number of authors are readily available, see, e.g., [5] and [9]. Though *polynomial models* are attractive in many applications [such as to coding theory], they may not be particularly convenient when studying certain classes of problems in system theory and digital filtering. In these areas we are often able to identify a transfer function $f(x)$ of a certain linear time-invariant system $u(z) \rightarrow \boxed{f(z)} \rightarrow y(z)$ via a finite number of interpolation conditions, and it is often advantageous to devise an efficient and robust procedure to construct an interpolant $f(x)$ belonging to a prescribed class of functions. For physical reasons this transfer function has to be passive, i.e., $|f(x)| \leq 1$ [to guarantee that the energy of the output, $y(z) = f(z)u(z)$, will not exceed the energy of the input $u(z)$]. Since polynomials are not bounded at infinity, one has to go beyond this class when studying many classes of applied problems. These arguments, among others, motivated many authors to study a variety of *rational* interpolation problems, first of all those related to the classical Nevanlinna-Pick problem recalled next.

1.2. Formulation: the classical scalar Nevanlinna-Pick problem. In [34] Nevanlinna proposed an efficient recursive algorithm to compute the solution for the following interpolation problem.

<u>Problem 1. Scalar Nevanlinna-Pick problem</u>.

<u>Given</u>: n points $\{z_k\}$ in the open right-half-plane Π^+, and n points $\{f_k\}$ inside the unit circle $\mathcal{D}$, i.e., $|f_k| < 1$.

<u>Construct</u>: a rational scalar function $f(x)$ such that

1. $f(z)$ is *analytic* inside Π^+.
2. $f(z)$ is *passive*: $\sup_{z \in \Pi^+ \cup i\mathbf{R}} |f(z)| \leq 1.$
3. $f(z)$ meets the interpolation conditions

$$f(z_k) = f_k \qquad \text{for } k = 1, 2, \ldots, n.$$

The well-known solvability condition [36] is that the corresponding Pick matrix,

$$R = \left[\frac{1 - f_i f_j^*}{z_i + z_j^*} \right] \tag{1}$$

should be positive definite.

1.3. Generalizations and applications. A strong interest has been brought to this problem, extending considerably its original settings, and putting it into the light of various relationships between seemingly unrelated abstract and applied areas.

To begin with, there is a vast operator theory literature on far reaching generalizations of the Nevanlinna-Pick problem. We only mention that deep results were obtained in the frameworks of several approaches and *"languages"*, including the band extension method by H.Dym and I.Gohberg, the Buerling-Lax-theorem approach by J.Ball and J.Helton, the state-space approach by K.Glover and J.Ball, I.Gohberg, L.Rodman, the de Branges reproducing kernel method by H.Dym, lifting-of-commutants method by C.Foias. We refer to [17], [12], [8], [18] for a discussion, as well as for a wide list of references.

Second, along with the interest of operator theorists, a growing attention to the Nevanlinna-Pick problem has been paid in the engineering community. As was noted above, one motivation to study of this class of interpolation problems comes from looking at $f(z)$ as at a transfer function of a certain linear time-invariant system, where the passivity is naturally imposed by the conservation of energy, and since we consider a finite number of interpolation conditions, it makes sense to look for rational interpolants. Thus, it is not surprising that many engineering applications of this interpolation problem have been discovered. Several such applications in system, control and circuit theory were surveyed in [14]; in particular, it has been observed that the Darlington synthesis procedure [11], well-known in network theory, is nothing else but a form of the classical Nevanlinna recursion. See also [2], [15], [25] for applications.

To sum up, the early results obtained by Pick and Nevanlinna about six decades ago have been found to be deep; and further studies mainly progressed in the two following directions: (i) far reaching operator-theory generalizations, and (ii) application of these to various electrical engineering problems. In this paper we address a different problem which does not seem to fall into either one of the above two categories: we study the *arithmetic complexity* of the Nevanlinna-Pick interpolation problem.

1.4. Arithmetic complexity of computations. The question is: given interpolation data at n points, $\{z_k, f_k\}_{1 \leq k \leq n}$, how many arithmetic operations we have to perform to compute the solution? The original Nevanlinna algorithm uses $O(n^2)$ operations. Before presenting our improvements, we next briefly clarify why this was already a very satisfactory result. It is well-known that the classical Nevanlinna algorithm can be seen as a way of speeding up computing the Cholesky factorization,

$$R = LL^* \qquad [\ L \text{ is lower triangular }]$$

for the corresponding Pick matrix (1). So, the arithmetic complexity $O(n^2)$ operations of the Nevanlinna algorithm compares favorably with the order-of-magnitude-higher complexity $O(n^3)$ of the standard [i.e., structure-ignoring] Cholesky factorization algorithm. Therefore, the Nevanlinna algorithm belongs to the class of what is now called *fast algorithms*. Can one do faster than $O(n^2)$?

1.5. Main result: a new superfast algorithm for the tangential boundary Nevanlinna-Pick problem. The main result of this paper is a new *superfast* algorithm for solving the more general *tangential* Nevanlinna-Pick problem [the name "tangential" was suggested by M.G.Krein]. The practical motivation to study such a more general case [where we seek a passive rectangular $M \times N$ rational matrix function $F(z) = \left[\frac{p_{ij}(z)}{q_{ij}(z)} \right]$] emerges from the multi-channel case.

Problem 2. Tangential Nevanlinna-Pick problem.

Given :

n distinct points $\{z_k\}$ in the open right-half-plane Π^+,
n nonzero row $N \times 1$ vectors $\{x_k\}$,
n row $M \times 1$ vectors $\{y_k\}$.

Construct: a rational $N \times M$ matrix function $F(x)$ s.t.

1. $F(z)$ is *analytic* inside Π^+.
2. $F(z)$ is *passive*: $\sup_{z \in \Pi^+ \cup i\mathbf{R}} \|F(z)\| \leq 1.$
3. $F(z)$ meets the *tangential* interpolation conditions $x_k \cdot F(z_k) = y_k$ for $k = 1, 2, ..., n.$

The well-known solvability condition is that the following $n \times n$ generalization of the Pick matrix,

$$R = \left[\frac{x_i x_j^* - y_i y_j^*}{z_i + z_j^*} \right] \tag{2}$$

is positive definite. Solutions for this *tangential* variant of the problem that lead to *fast* $O(n^2)$ algorithms can be found in several places [see, e.g., [29], [16], [13], [12], [20], [40], [21], [27] among others]. Here we present a new *superfast* $O(n \log^3 n)$ algorithm.

In fact, this algorithm solves several more rational interpolation problems including the following more general variant of Problem 2, in which the first m $(< n)$ points $\{z_k\}$ lie on the boundary $i\mathbf{R}$ of Π^+.

Problem 3. Tangential boundary Nevanlinna-Pick problem.

Given: The same data $\{z_k, x_k, y_k\}$ as in Problem 2, but now

- the first m $(< n)$ points $\{z_k\}$ lie on the imaginary line $i\mathbf{R}$.
- $\|x_k\| = \|y_k\|$ for $k = 1, 2, ..., m$.
- We are also given m positive numers $\{\rho_k\}$ [called *coupling numbers*].

Construct: a rational $N \times M$ matrix function $F(x)$ satisfying conditions 1)-3)of Problem 2 and also:

4. $x_k \cdot F'(z_k) \cdot y_k^* = -\rho_k$ for $k = 1, 2, ..., m$.

The known condition [8] for solvability of Problem 3 is that the following $n \times n$ generalization of Pick matrix

$$R = \left[\begin{array}{c} r_{ij} \end{array} \right] \quad \text{with} \quad r_{ij} = \begin{cases} \rho_i & \text{if } i = j \le m \\ \frac{x_i x_j^* - y_i y_j^*}{z_i + z_j^*} & \text{otherwise} \end{cases} \tag{3}$$

is positive definite.

1.6. Superfast algorithms for other interpolation and engineering problems. In fact, the main algorithm obtained here applies to several rational interpolation problem of Nevanlinna-Pick type, including the tangential Hermite-Fejer, matrix Nehari and Nehari-Takagi interpolation problems, as well as to several engineering problems, including model reduction, sensitivity minimization, and robust stabilization. The details on these interpolation problems will be described in the full paper; here we only briefly indicate how our new algorithm solves the following problem.

Problem 4. Suboptimal matrix Nehari problem.

Given :

n distinct points $\{z_k\}$ in the open left-half-plane Π^-,
n $1 \times N$ row vectors $\{w_k\}$,
n $M \times 1$ column vectors $\{\gamma_k\}$,

and a rational $M \times N$ matrix function $K(z)$ having only n simple poles $\{z_k\}$ in Π^-, such that

$$K(z) = (z - z_k)^{-1} \gamma_k \cdot w_k + \text{ [analytic at } z_k] \,,$$

and such that $\sup_{z \in i\mathbf{R}} \|K(z)\| < \infty$.

Construct: A rational matrix function $R(z)$ with no poles in $\Pi^- \cup i\mathbf{R}$, such that $\sup_{z \in i\mathbf{R}} \|K(z) - R(z)\| < 1$.

The solvability condition [8] is that the $2n \times 2n$ matrix

$$R = \left[\begin{array}{cc} Q & I \\ I & P \end{array} \right] \tag{4}$$

has exactly n positive and n negative eigenvalues, where

$$P = \left[\begin{array}{c} -\frac{w_i \cdot w_j^*}{z_i + z_j^*} \end{array} \right], \qquad Q = \left[\begin{array}{c} -\frac{\gamma_i^* \cdot \gamma_j}{z_i^* + z_j} \end{array} \right].$$

1.7. Superfast algorithms for generalized Pick matrices. As we mentioned in Sec. 1.4, the classical Nevanlinna algorithm can be seen as a fast $O(n^2)$ way of computing the Cholesky factorization for the corresponding Pick matrix (1). Similarly, our new algorithm can also be seen as a superfast $O(n \log^3 n)$ method for triangular factorization and inversion of generalized Pick matrices in (2) and (3), and for the other important classes of structured matrices defined next.

1.8. Displacement structure. In devising the algorithm we exploit an immediately verified but fruitful observation that the generalized Pick matrices in (2) and (3) satisfy the equation

$$A_\pi^* R + R A_\pi = -C_\pi^* \left[\begin{array}{cc} I_N & 0 \\ 0 & -I_M \end{array} \right] C_\pi. \tag{5}$$

where

$$A_\pi = \text{ diag } (-z_1^*, \ldots, -z_n^*), \quad C_\pi = \left[\begin{array}{ccc} -x_1^* & \cdots & -x_n^* \\ -y_1^* & \cdots & -y_n^* \end{array} \right].$$

The important fact that will be used and extended in a moment is that (5) implies that the number

$$\text{rank } (A_\pi^* R + R A_\pi) = M + N \tag{6}$$

can be much smaller than the size n of R [for example, $M + N = 2$ for the usual Pick matrices (1)]. In the interpolation context, this means that the size of an $M \times N$ interpolant $F(z)$ of Problems 1)-4) is usually much smaller then the number n of interpolation points $\{z_k\}$.

Many applications give rise to various other kinds of large dense *structured* matrices. Similarly to the Pick matrices, their structure is understood in the sense that their n^2 entries are defined by a much smaller number $O(n)$ of parameters. Table 1 presents some examples.

Table 1. Examples of matrices with structure.

Toeplitz, $T = [\ t_{i-j}\ ]$	Hankel, $H = [\ h_{i+j}\ ]$	Vandermonde, $V = [\ x_i^{j-1}\ ]$
Chebyshev-Vandermonde, $V_T = [\ T_{j-1}(x_i)\ ]$ $T_k(x)$ are Cheb. pol.	Cauchy, $C = \left[\ \frac{1}{x_i - y_j}\ \right]$	Bezoutians, controllability, observabilty matrices, etc.

Similarly to Pick matrices in (6), for each of these patternts of structure one can choose matrices $\{A_\pi, A_\zeta\}$ so that the number

$$\alpha(R) = \text{ rank } (RA_\pi - A_\zeta R) \tag{7}$$

is small. Table 2 lists some choices for $\{A_\pi, A_\zeta\}$.

Table 2. Displacement structure of matrices in Table 1

R	A_π	A_ζ	rank $(RA_\pi - A_\zeta R)$
Toeplitz	Z	Z	2
Hankel	Z^T	Z	2
transp. Vandermonde	Dx	Z	1
Cauchy	D_y	D_x	1

Here Z is the lower shift matrix, having 1 on the first subdiagonal and zeros elsewhere, and $D_x = \text{diag}(\text{x}_1, \cdots, \text{x}_\text{n})$.

The number $\alpha(R)$ in (7) is called the *displacement rank* of R, and if the displacement rank is small then R is said to have a *displacement structure*. The [easily verified] facts in Table 2 show that the matrices listed in Table 1 all have displacement structure.

Many applications, however, go beoynd the simplest examples of Table 1, and give rise to the more general classes of matrices with displacement structure. These are matrices with small displacement rank [i.e., $\alpha(R)$ can be slightly bigger than just 1 or 2 shown in Table 2]. For example, Cauchy-like matrices are defined as those with the small $\alpha(R) = \text{ rank } (D_x R - RD_y)$. According to this definition, Pick matrices are Cauchy-like, because $\{A_\pi^*, A_\pi\}$ in (5) are diaginal. Similarly, the other choices for $\{A_\pi, A_\zeta\}$ in Table 2 lead to the definitions of Toeplitz-like, Hankel-like and Vandermonde-like matrices.

There is a nice theory of such matrices with displacement structure[1]. We refer to surveys [24], [30], [35] for various aspects of displacement and further references.

1.9. Superfast algorithms for matrices with [partially reconstructible] displacement structure. Fast $O(n^2)$ and superfast $O(n \log^2 n)$ algorithms for trianguar factorization were first designed for Toeplitz-like matrices in [32], [33], [6]. Following the breakthrough work of M.Morf, fast $O(n^2)$ algorithms were exteded to the other types of displacement structure including Vandermonde-like, Cauchy-like, Hankel-like matrices and others, and several techniques to design such fast algorithms can be found, e.g., in [24], [30], [21], [35] among others. However, the superfast algorithm of [33], [6] for Toeplitz-like matrices has not being carried over to all other types of matrices with displacement structure. This gap is filled by our new superfast algorithm. Though it is designed here for solving various tangential interpolation problems, it can also be used for computing traingular factorization and for inversion of matrices with various displacement structures including Cauchy-like, Vandermonde-like and Hankel-like matrices [even for Toeplitz-like matrices it offers an improvement over the MBA algorithm of [33], [6]].

Finally, we note that *boundary* tangential interpolation problems [see, e.g., Problem 3], and Nehari problem in fact give rise to what we called in [27] *partially reconstructible* matrices. We recall the formal definitions in the main text below, here we only mention that the new superfast algorithm applies to this more general displacement structure as well.

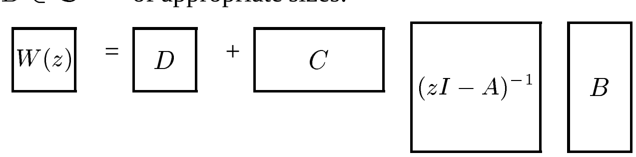
1.10. Contents. The paper is structured as follows. In the section 2 we briefly recall a global state-space formula for solutions of the boundary tangential Nevanlinna-Pick problem. This nice formula has theoretical significance, but it does not immediately lead to fast and superfast algorithms. In order to design a superfast algorithm in section 4, we first describe in section 3 factorization theorems, that led in [20], [21] to fast $O(n^2)$ algorithms. In section 4 we use a divide-and-conquer approach to factorization, obtaining a new superfast $O(n \log^3 n)$ algorithm for the boundary tangential Nevanlinna-Pick problem. In section 5 we briefly indicate another important rational interpolation problem, the matrix Nehari problem, to show that the algorithm solves the other interpolation problems of the Nevanlinna-Pick type. Some conclusions are offered in the last section.

2. State-space approach to the tangential Nevanlinna-Pick problem

2.1. State-space approach. In the early 1960's R.E.Kalman introduced a state-space method to study a linear time-invariant dynamical systems. This method is based on a representation of the transfer function for such systems, i.e., a rational $m \times m$ matrix function $W(z) = \left[\begin{array}{c} \frac{p_{ij}(z)}{q_{ij}(z)} \end{array} \right]$, in the form called a *realization*,

$$W(z) = D + C(zI - A)^{-1}B, \tag{8}$$

with the matrices $D \in \mathbf{C}^{\alpha\times\alpha}$, $C \in \mathbf{C}^{\alpha\times n}$, $A \in \mathbf{C}^{n\times n}$, $B \in \mathbf{C}^{n\times\alpha}$ of appropriate sizes:



A realization is called *minimal*, if the size of A is the smallest possible. The attractive feature of this approach is in that the representation (8) often reduces a non-linear problem to a simpler linear algebra problem, involving just four arrays in $\{A, B, C, D\}$.

[1]The nomenclature "*displacement*" was suggested in [19], [26] because these references first explored the approach to study Toeplitz-like matrices defined using shift [= displacement] matrices Z, cf. with Table 2.

Although the original interest in the state-space method was for specific engineering applications [e.g., the linear quadratic regulator problem and the Kalman filter], the state-space method turned out to be fundamental, leading to new insights in many other directions. Various engineering and mathematical problems have been addressed in this way, see, e.g., [17], [15], [2], [8] among many others. However, it has been found [8] that solutions for all these tangential interpolation problems can be more conveniently described via a special *global state-space* representation for the interpolant,

$$W(z) = I + C_\pi (zI - A_\pi)^{-1} R^{-1} B_\zeta, \tag{9}$$

now involving [as opposed to (8)] the inverse of a certain structured matrix R, satisfying the following matrix equation

$$RA_\pi - RA_\zeta = B_\zeta C_\pi. \tag{10}$$

Such equations are often called *displacement equations*, and such R are often referred to as matrices with *displacement structure*, see, e.g., Sec. 1.8.

This connection [8] of interpolants $W(z)$ in (9) to matrices R with displacement structure in (10) has already been exploited to devise fast algorithms for rational interpolation [20], [40], [21], [35], and here we continue this work, and design a superfast algorithm for the more involved boundary problems.

2.2. A solution to the boundary tangential Nevanlinna-Pick problem. Before describing our algorithm we first present the *global state-space* solution of the form (9) for the boundary tangential Nevanlinna-Pick problem. It turns out that the set of all solutions can be parameterized by the formula (9) as follows [cf. with [8]].

- Use the given data of Problem 3 of Sec. 1.5 to form the following matrices:

$$C_\pi = \begin{bmatrix} -x_1^* & \cdots & -x_n^* \\ -y_1^* & \cdots & -y_n^* \end{bmatrix}, B_\zeta = \begin{bmatrix} x_1 & -y_1 \\ \vdots & \vdots \\ x_n & -y_n \end{bmatrix},$$

  $A_\pi = \operatorname{diag}(-z_1^*, \ldots, -z_n^*)$, and form R to be the generalized Pick matrix displayed in (3).

- Use these matrices $\{C_\pi, A_\pi, B_\zeta, R\}$ to write down an $(M+N) \times (M+N)$ rational matrix function of the form (9), and partition it as

$$W(z) = \begin{bmatrix} W_{11}(z) & W_{12}(z) \\ W_{21}(z) & W_{22}(z) \end{bmatrix}.$$

- Then all solutions $F(z)$ for the boundary tangential Nevanlinna-Pick problem are parametrized as follwos

$$F(z) = [W_{11} \cdot G(z) + W_{12}] \cdot [W_{21} \cdot G(z) + W_{22}]^{-1}, \tag{11}$$

  where $G(z)$ is an arbitrary rational matrix function satisfying 1) and 2) of problem 2 in Sec 1.5 (with G in place of F).

2.3. Cascade decompostion. Formulas (11), (9) present a nice closed-form expression for solutions without any computations. Unfortunately, computations are needed when these two formulas are used to evaluate the interpolant $F(z)$, in this case the inversion of R in (9) [or solving the associated linear systems] becomes a computational and numerical bottleneck. If standard structure-ignoring methods [such as Cholesky, QR factorization, bordering method, etc.] are employed for this purpose, then the resulting complexity would be too expensive: $O(n^3)$ arithmetic operations. This motivates one to look for an alternative to (9) representation of $W(z)$. One such more convenient representation is a cascade decomposition of $W(z)$,

$$W(z) = \Theta_1(z) \cdot \ldots \cdot \Theta_n(z) \tag{12}$$

where

$$\Theta_k(z) = I + c_k \frac{1}{z - z_k} \frac{1}{d_k} b_k$$

are the *first order* factors, i.e., they have the same form (9), but they have an advantage of having just one pole z_k and one zero $-z_k^*$. Since c_k is just a $\alpha \times 1$ column, d_k is a scalar, and b_k is just a $1 \times \alpha$ row, the representation (12) is very attractive, because it allows us to evaluate $W(z)$ in only $O(\alpha n)$ operations. Another argument in favor of (12) is that such first-order sections $\Theta_k(z)$ are easier to realize as electronic devices, so that by concatenating these sections in a cascade one obtains a realization for the global transfer function $W(z)$.

In this paper we devise a superfast algorithm for *factorization of rational matrix functions*. Using the formulas (11), (9) as an input, our algorithm computes the solution $W(z)$ for the boundary tangential Nevanlinna-Pick problem in the attractive form (12). Similarly, it can compute solutions for the other rational interpolation problems of this kind, e.g., for matrix Nehari, Nehari-Takagi and Caratheodory-Fejer problems.

3. Factorization of rational matrix functions and displacement structure

3.1. Interpolants and displacement. There is a large number of theorems, e.g., in [37], [39], [7], [23], [3], [4], [31], [20], [40], [21] on factorizations of rational matrix functions. In order to design a superfast factorization algorithm we recall in the next two sections a variant of [21] of such a theorem, and refer to a recent survey [35] that discusses it in a historical context and lists some applications.

We reinforse the point made at the end of Sec. 2.1 that there is a close connection between interpolants $W(z)$ of the form (9) [that we are going to factorize] and matrices R with dispacement structure in (10). Keeping this connection in mind, we postpone presenting the factorization result for $W(z)$ till Sec. 3.3, and first present in Sec. 3.2 a purely matrix auxiliary lemma for R with displacement structure.

3.2. Schur complements of structured matrices. Our notations here involve the same subscripts as in (9) and (10), to keep notations unified throughout the paper.

Lemma 3.1 [[20], [21], Theorem 2.3] *Let matrices in*

$$R_1 A_\pi^{(1)} - A_\zeta^{(1)} R_1 = B_\zeta^{(1)} C_\pi^{(1)} \tag{13}$$

be partitioned as $R_1 = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$,

$$A_\pi^{(1)} = \begin{bmatrix} A_{\pi,1}^{(1)} & * \\ 0 & A_{\pi,2}^{(1)} \end{bmatrix}, \quad A_\zeta^{(1)} = \begin{bmatrix} A_{\zeta,1}^{(1)} & 0 \\ * & A_{\zeta,2}^{(1)} \end{bmatrix}, \tag{14}$$
$$C_\pi^{(1)} = \begin{bmatrix} C_{\pi,1}^{(1)}, & C_{\pi,2}^{(1)} \end{bmatrix}, \quad B_\zeta^{(1)} = \begin{bmatrix} B_{\zeta,1}^{(1)} \\ B_{\zeta,2}^{(1)} \end{bmatrix},$$

If R_{11} is nonsingular then the Schur complement

$$R_2 = R_{22} - R_{21} R_{11}^{-1} R_{12}$$

satisfies the equation

$$R_2 A_\pi^{(2)} - A_\zeta^{(2)} R_2 = B_\zeta^{(2)} C_\pi^{(2)} \tag{15}$$

where $A_\pi^{(2)} := A_{\pi,2}^{(1)}$, $A_\zeta^{(2)} := A_{\zeta,2}^{(1)}$ *are just "borrowed" from (14), and*

$$C_\pi^{(2)} = C_{\pi,2}^{(1)} - C_{\pi,1}^{(1)} R_{11}^{-1} R_{12}, \quad B_\zeta^{(2)} = B_{\zeta,1}^{(1)} - R_{21} R_{11}^{-1} B_{\zeta,2}^{(1)}. \tag{16}$$

Recall that in applications structured matrices usually appear *implicitly*, i.e., their n^2 entries are defined by a smaller number $O(n)$ of parameters. For example, in Problem 2 we are given $(1+M+N)\times n$ parameters in $\{z_k, x_k, y_k\}_{1\le k\le n}$, and *not* the entries of the generalized Pick matrix of (2).

The whole point of a *displacement approach* to structured matrices is in avoiding manipulations on the entries, and instead working with a small number of parameters. Assuming for the moment that R_1 is a unique solution of the equation (13), we see that the whole information on R_1 is conveniently captured by the four *generator* matrices in

$$\{C_\pi^{(1)}, A_\pi^{(1)}, A_\zeta^{(1)}, B_\zeta^{(1)}\} \tag{17}$$

which usually involve only $O(n)$ parameters, cf., e.g., with Table 2. For example, if R_1 is the generalized Pick matrix of (2), (5), then $\{A_\pi, A_\zeta\}$ are diagonal, C_π has only α rows, and B_ζ has only α columns [with $\alpha << n$].

The above discussion shows that Lemma 3.1 allows one to replace expensive computing the Schur complement

$$R_1 \longrightarrow R_2$$

by a much cheaper computing via (16) its generator,

$$\{C_\pi^{(1)}, A_\pi^{(1)}, A_\zeta^{(1)}, B_\zeta^{(1)}\} \longrightarrow \{C_\pi^{(2)}, A_\pi^{(2)}, A_\zeta^{(2)}, B_\zeta^{(2)}\}. \tag{18}$$

However, Lemma 3.1 allows us to treat the more general situation addressed next.

3.3. Partially reconstructuble matrices. We note two properties of Lemma 3.1 that will be crucial in Sec. 4 for the design of a superfast algorithm.

Two properties.

1. Lemma 3.1 addresses a *block* Schur complementation. This is important, because the usual technique of devising superfast algorithms, *divide-and-conquer*, is based on partioning R_1 into four nearly equal *blocks*.

2. Lemma 3.1 does not assume that the displacement equation (13) has a *unique* solution R_1. This is important, because for *boundary* interpolation problems [and for some other, e.g., for the Nehari problem] the associated displacement equations have many solutions. For example, the generalized Pick matrix of (3), is a *nonunique* solution of (5). Indeed, the first m diagonal entries of R [coupling numbers $\{\rho_k\}$] cannot be recovered from (5).

The second property shows that in order to design an algorithm, we need to modify the recursion in (18), because a generator of R_1 in (17) no longer contains the full information on R_1.

Denote by $\mathcal{K} = \mathrm{Ker}\nabla_{\{A_\pi^{(1)}, A_\zeta^{(1)}\}}(\cdot)$ the kernel of the *displacement operator*, $\nabla_{\{A_\pi^{(1)}, A_\zeta^{(1)}\}}(\cdot) : \mathbf{C}^{n\times n} \to \mathbf{C}^{n\times n}$,

$$\nabla_{\{A_\pi^{(1)}, A_\zeta^{(1)}\}}(R_1) = R_1 A_\pi^{(1)} - A_\zeta^{(1)} R_1. \tag{19}$$

If $\mathcal{K}$ is non-trivial, we call such R_1 *partially reconstructible*, because now its generator (17) no longer contains the full information on R_1. Following [27], let us represent R_1 as

$$R_1 = R_{\mathcal{K}}^{(1)} + R_{\mathcal{K}^\perp}^{(1)} \quad \text{with respect to} \quad \mathbf{C}^{n\times n} = \mathcal{K} \oplus \mathcal{K}^\perp, \tag{20}$$

where the orthogonality of matrices is understood in the sense of inner product

$$< A, B >= \mathrm{tr}(B^* \cdot A), \tag{21}$$

where $\mathrm{tr}(A)$ denotes the sum of all diagonal entries of A.

A partially reconstructible matrix R_1 is uniquely determined by the five matrices

$$\{C_\pi^{(1)}, A_\pi^{(1)}, A_\zeta^{(1)}, B_\zeta^{(1)}, R_{\mathcal{K}}^{(1)}\}, \quad (22)$$

so we called them a *generator* of R_1, see [27].

It is easy to see that for all basic classes of partially reconstructible structured matrices [i.e., Toeplitz-like, Hankel-like, Cauchy-like, etc.], the matrix $R_{\mathcal{K}}^{(1)}$ is defined by a small number $m(\leq n)$ of parameters. For example, for generalized Pick matrices in (3), (5) we have $R_{\mathcal{K}}^{(1)} = \text{diag}\,(\rho_1, \ldots, \rho_m, 0, \ldots, 0)$.

3.4. Factorization of rational matrix functions. In fact, Lemma 3.1 was obtained as a matrix interpretation of the following factorization result.

Theorem 3.2 [[20], [21], Theorem 2.3] *Let matrices* $\{C_\pi, A_\pi, A_\zeta, B_\zeta, R_1\}$ *be defined as in Lemma 3.1 [e.g., satisfying (13)]. If the left upper block* R_{11} *is invertible then the rational matrix function*

$$W_1(z) = I + C_\pi^{(1)}(zI - A_\pi^{(1)})^{-1} R_1^{-1} B_\zeta^{(1)},$$

admits a minimal factorization [minimal means that there is no pole-zero cancelation]

$$W_1(z) = W_{2,1}(z) \cdot W_{2,2}(z),$$

where the first factor

$$W_{2,1}(z) = I_p + C_{\pi,1}^{(1)} \cdot (zI_{N_1} - A_{\pi,1}^{(1)})^{-1} \cdot R_{11}^{-1} \cdot B_{\zeta,1}^{(1)}, \quad (23)$$

is obtained "by extraction", i.e., without computations, and the second factor

$$W_{2,2}(z) = I_P + C_\pi^{(2)}(zI - A_\pi^{(2)})^{-1} R_2^{-1} B_\zeta^{(2)}$$

involves Schur complement R_2 *and matrices* $\{C_\pi^{(2)}, B_\zeta^{(2)}\}$ *computed by (16).*

Theorem 3.2 shows that factorization of a rational matrix function $W_1(z)$ can be computed via computing Schur complement for R_1 based on Lemma 3.1. In [20], [21] this technique has been exploited to design a fast $O(n^2)$ factorization algorithm. In the next section we further accelerate the factorization and design a superfast algorithm.

4. Divide-and-conquer approach and a new superfast algorithm

4.1. Preliminary observations. Here we use the results of Sec. 3 to derive the main algorithm of the paper, and this will be done in two steps. For simplicity we first describe in Sec 4.2 a superfast implementation of Lemma 3.1 only, because it allows us to use purely matrix arguments. Having this background, we then not only show in Sec. 4.3 that it solves the boundary tangential Nevanlinna-Pick interpolation problem, but we also provide a clarifying interpolation interpretation for each step of the algorithm.

4.2. Matrix description of the main algorithm. Applying a displacement approach described in Sec. 3.2 and 3.3, we avoid computations on the entries of R_1 and manipulate instead on its generator in (22) in the following two fundamental computations.

- Instead of computing the entries of the Schur complement R_2 we use Lemma 3.1 to compute its generator.
- The previous item uses the fact [see Lemma 3.1] that R_2 inherits the structure of R_1. It is known and can be easily seen that the inverse of R_1 in (13) inherits its structure as well:

  $$A_\pi^{(1)} R_1^{-1} - R_1^{-1} A_\zeta^{(1)} = (R_1^{-1} B_\zeta^{(1)})(C_\pi^{(1)} R_1^{-1}). \quad (24)$$

  Therefore, instead of computing the entries of R_1^{-1}, we shall compute its generator.

<u>Procedure D&C-Factor</u> [Matrix formulation]

<u>Input</u>: A generator (22) of R_1 in (13).

<u>Output</u>:

1. A generator for the Schur complement $R_2 = R_{22} - R_{21}R_{11}^{-1}R_{12}$.
2. A generator for the inverse R_1^{-1}.

<u>Steps</u>:

1. Consider a partition $R_1 = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$ into four matrices of nearly equal sizes. Extract by inspection from (13) generators for submatrices R_{11}, R_{12}, R_{21}.
2. Use a generator for R_{11} and apply the algorithm D&C-Factor to R_{11} to obtain a generator for its inverse R_{11}^{-1}.
3. Compute a generator for the Schur complement R_2 using the formulas of Lemma 3.1:

   $$C_\pi^{(2)} = C_{\pi,2}^{(1)} - C_{\pi,1}^{(1)} R_{11}^{-1} R_{12},$$

   $$B_\zeta^{(2)} = B_{\zeta,1}^{(1)} - R_{21} R_{11}^{-1} B_{\zeta,2}^{(1)}.$$

   and compute $R_{\mathcal{K}}^{(2)}$ by using the standard Schur complementation formula[2].

[2]The Schur complementation formula $R_2 = R_{22} - R_{21}R_{11}^{-1}R_{12}$ is used here to compute only a *few numbers* defining $R_{\mathcal{K}}^{(2)}$, e.g., coupling numbers $\{\rho_k\}$ in case of the generalized Pick matrix (3).

4. Use this generator of R_2 to apply the algorithm D&C-Factor to R_2 to obtain the generator for its inverse R_2^{-1}.

5. Use generators for $R_{11}^{-1}, R_2^{-1}, R_{12}, R_{21}$ to compute the generator for R_1^{-1} by using the representation

$$R^{-1} =$$

$$\left[\begin{array}{cc} I & -R_{11}^{-1}R_{12} \\ 0 & I \end{array}\right]\left[\begin{array}{cc} R_{11}^{-1} & 0 \\ 0 & R_2^{-1} \end{array}\right]\left[\begin{array}{cc} I & 0 \\ -R_{21}R_{11}^{-1} & I \end{array}\right]$$

for fast multiplcation of matrices on the right-hand side of (24).

4.3. Analysis of complexity. Let us denote by $C(n)$ the arithmetic complexity of the algorithm D&C-Factor for an $n \times n$ matrix R_1, and by $M(n)$ complexity of multiplication by a vector for a class of structured matrices[3] including R_1, its submatrices, Schur complements and inverses [we shall be more specific regarding these classes in a moment]. Then the analysis of the algorithm D&C-Factor results in the operations count

$$C(n) = 2[2C(\frac{n}{2}) + 8M(\frac{n}{2}) + O(n)].$$

Therefore the arithmetic complexity of the D&C-Factor algorithm depends upon the estimate $M(n)$. If $M(n) = O(n \log^c n)$ for some nonnegative constant c, then $C(n) = O(n \log^{1+c} n)$. In the next Table we spesify this result for several particular patterns of structure, and refer to [22] for the estimates for $M(n)$.

Table 3. Complexity.

	$M(n)$	$C(n)$
Toeplitz-like	$O(n \log n)$	$O(n \log^2 n)$
Hankel-like	$O(n \log n)$	$O(n \log^2 n)$
Vandermonde-like	$O(n \log^2 n)$	$O(n \log^3 n)$
Cauchy-like	$O(n \log^2 n)$	$O(n \log^3 n)$

We finally note that for Cauchy-like and Vandermonde-like matrices with special nodes, such as roots of unity, or Chebyshev points, the algorithm admits further acceleration by using FFT or FCT/FST.

4.4. Interpolation interpretation and a solution for the boundary tangential Nevanlinna-Pick problem.

The algorithm D&C-Factor was discussed so far in purely matrix framework, but let us return to the interplay between Lemma 3.1 and Theorem 3.2. Now we are able to present the results of the analysis that shows that the D&C-Factor algorithm in fact computes a minimal factorization

$$W_1(z) = W_{2,1}(z)W_{2,2}(z),$$

and that this algorithm calls itself to further recursively factorize each of these two factors. Thus one more output of this algorithm is the desired [see, e.g., Sec. 2.3] complete cascade decomposition

$$W_1(z) = \Theta_1(z) \cdot \ldots \cdot \Theta_n(z)$$

into a product of first order factors. As we noted in Sec. 2.3 and 3.4, this solves the boundary tangential Nevanlinna-Pick problem, and the achieved arithmetic complexity $O(n \log^3 n)$ dramatically improves the best known bound $O(n^2)$ for rational interpolation problems of this kind.

Procedure D&C-Factor [Interpolation formulation]

Input: A rational matrix function

$$W_1(z) = I + C_\pi^{(1)}(zI - A_\pi)^{-1}R_1^{-1}B_\zeta^{(1)}$$

Output:

1. A minimal factorization

$$W_1(z) = W_{2,1}(z)W_{2,2},$$

2. The inverse

$$W_1^{-1}(z) = I + C_\zeta(zI - A_\zeta)^{-1}RB_\pi.$$

Steps:

1. Write down the first factor

$$W_{2,1}(z) = I_p + C_{\pi,1}^{(1)} \cdot (zI_{N_1} - A_{\pi,1}^{(1)})^{-1} \cdot R_{11}^{-1} \cdot B_{\zeta,1}^{(1)}.$$

2. Apply the algorithm D&C-Factor to $W_{2,1}(z)$ to obtain its inverse $W_{2,1}^{-1}(z)$.
3. Compute the factor $W_{2,2}(z)$ via (16), as suggested by Theorem 3.2. and Lemma 3.1.
4. Apply the algorithm D&C-Factor to $W_{2,2}(z)$ to obtain its inverse $W_{2,2}^{-1}(z)$.
5. Multiply the factors in

$$W_1^{-1}(z) = W_{2,2}^{-1}(z) \cdot W_{2,1}^{-1}(z).$$

Finally we observe, that we presented a general version of the divide-and-conqure factorization algorithm that admits further computational improvements for important cases. For example, for the boundary tangential Nevanlinna-Pick problem, the genaralized Pick matrix (3) is Hermitian. By exploiting its symmetry we are able to further reduce the amount of computations by the factor 2.

If we have special interpolation points, e.g., roots of unity or Chebychev points, then the algorithm can be further accelerated via FFT, FCT or FST. Moreover, the numerical properties of the suggested algorithm require further investigation,and they depend on the configuration of the interpolation points.

[3]Once again, such fast multiplication $R \cdot b$ should avoid operations on the entries of R and it should use its generator.

5. Matrix Nehari problem

The new D&C-Factor algorithm can be applied to factorize any rational matrix function of the *global state-space* form (9). Recently such global state-space solutions have been obtained for a number of interpolation problems and engineering problems, e.g., mentioned in Sec. 1.6. We shall provide more details on these problems in the full paper, and here only briefly indicate how the D&C algorithm improves the earlier estimate $O(n^2)$ operations obtained for Problem 4 in [20], [21]. This follows from the following formulas that should be used as an input for our D&C algorithm. The solution $F(z) = K(z) - R(z)$ for Problem 4 was parametrized in [8] by the same formula (11) with an arbitrary parameter $G(z)$ satisfying **(i)** $G(z)$ has no poles in $\Pi^- \cup i\mathbf{R}$, **(ii)** $\sup_{z \in i\mathbf{R}} \|G(z)\| \leq 1$; and with $W(z)$ of the form (9) now involving the following matrices: $A_\pi = \text{diag}\,(z_1, \ldots, z_n, -z_1^*, \ldots, -z_n^*)$, $J = \begin{bmatrix} I_N & 0 \\ 0 & -I_M \end{bmatrix}$,

$$C_\pi = \begin{bmatrix} \gamma_1 & \ldots & \gamma_n & 0 & \ldots & 0 \\ 0 & \ldots & 0 & w_1^* & \ldots & w_n^* \end{bmatrix}, \quad B_\zeta = C_\pi^* \cdot J,$$

where partially reconstructible R in 4) satisfies

$$A_\pi R + R A_\pi^* = C_\pi^* J C_\pi.$$

6. Some concluding remarks

We present a new divide-and-conquer algorithm for factorization of rational matrix functions. The arithmetic complexity of the algorithm is $O(n \log^3 n)$ or less [e.g., $O(n \log^2 n)$ in some cases] which compares favorably with the much higher complexity $O(n^2)$ of earlier schemes. The input of our algorithm is a transfer function in the global state-space form. Since many classical rational interpolation problems and actual engineering problems were recently reduced to explicit state-space formulas, the new algorithm solves a variety of mathematical and important practical problems. Here we described an application of the algorithm to the boundary tangential Nevanlinna-Pick interpolation problem and the matrix Nehari problem. The details for other similar problems will appear in the full paper.

We specified the algorithm for "continuous-time" interpolation problems [i.e., with respect to the half-plane]. The corresponding "discrete-time" problems [i.e., with respect to the unit disk] can be solved similarly, see, e.g., [21], [27] for the corresponding transformation formulas and techniques.

The algorithm can be used for inversion and triangular factorization of matrices with displacement structure, including important classes of Toeplitz-like, Hankel-like, Vandermonde-like and Cauchy-like matrices.

References

[1] N.I.Akhiezer, *The classical moment problem and some related problems in analysis*, Hafner Publishing Co., New York, 1965.

[2] A.C.Antoulas, J.A.Ball, J.Kang, J.C.Willems, *On the solution of the minimal rational interpolation problem*, Linear Algebra Appl., **137-138** (1990) 479-509.

[3] D.Alpay and H.Dym, *On applications of reproducing kernel spaces to the Schur algorithm and rational J-unitary factorizations*, in I.Schur Methods in Operator Theory and Signal Processing (I.Gohberg, ed.), OT18, Birkhäuser Verlag, Basel, (1986) 89-160.

[4] D.Alpay and I.Gohberg, *Unitary rational matrix functions* in Topics in interpolation theory of rational matrix functions (I.Gohberg, ed.), OT33, Birkhäuser Verlag, Basel, (1988) 175-222.

[5] A.V. Aho, J.E.Hopcroft and J.D.Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, Mass., 1974.

[6] R.Bitmead and B.Anderson, *Assimptotically fast solution of Toeplitz and related systems of linear equations*, Linear Algebra and its Applications, bf 34 (1980) 103-116.

[7] H.Bart, I.Gohberg, M.A.Kaashoek and P.Van Dooren, *Factorizations of transfer functions*, SIAM J. Control and Optim., **18** (1980) 675-696.

[8] J. Ball, I.Gohberg and L.Rodman, *Interpolation of rational matrix functions*, OT45, Birkhäuser Verlag, Basel, 1990.

[9] A.Borodin and I.Munro, *The computational complexity of algebraic and numeric problems*, Amer. Elsvier, 1975.

[10] D.Bini and V.Pan, *Polynomial and vector computations. Vol. 1: Fundamental algorithms*, Birkhäuser, New York, 1994.

[11] S.Darlington, *Synthesis of reactance 4-poles which produce prescribed insertion loss characteristics*, J. Math. Physics, **18** (1939) 257-355.

[12] H.Dym, *J-contractive matrix functions, reproducing kernel Hilbert spaces and interpolation*, AMS, Providence, 1989.

[13] Ph.Delsarte, Y.Genin and Y.Kamp, *The Nevanlinna-Pick problem for matrix-valued functions*, SIAM J. Appl. Math., **36** (1979) 47-61.

[14] Ph.Delsarte, Y.Genin and Y.Kamp, *On the role of the Nevanlinna-Pick problem in circuit and system theory*, Circuit Theory and Appl., **9** (1981) 177-187.

[15] J.C.Doyle, K.Glover, P.Phargonekar, and B.A.Francis, *State-space solutions to standard H_2 and H_∞ problems*, IEEE Transactions on Automatic Control, **AC-34** (1989) 831-847.

[16] I.P.Fedchina, *Tangential Nevanlinne-Pick problem with multiple points*, Doklady Akad. Nauk Arm. SSR, **61** (1975) 214-218 (in Russian).

[17] C.Foias and A.E.Frazho, *The Commutant Lifting Approach to Interpolation Problems*, OT44, Birkhäuser, Basel, 1989.

[18] C.Foias, A.E.Frazho, I.Gohberg and M.A.Kaashoek, *Metric Constrained Interpolation, Commutant Lifting and Systems,* OT100, Birkhäuser, Basel, 1998.

[19] B.Friedlander, M.Morf, T.Kailath and L.Ljung, *New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices,* Linear Algebra and Appl., **27** (1979) 31-60.

[20] I.Gohberg and V.Olshevsky, *Fast algorithm for matrix Nehari problem,* Proceedings of MTNS-93, Systems and Networks: Mathematical Theory and Applications, v.2, Invited and Contributed Papers,edited by U. Helmke, R. Mennicken and J. Sauers, Academy Verlag, (1994) 687-690.

[21] I.Gohberg and V.Olshevsky, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations and Operator Theory, **20**, No. 1 (1994) 44-83.

[22] I.Gohberg and V.Olshevsky, *Complexity of multiplication with vectors for structured matrices*, Linear Algebra Appl., **202** (1994) 163-192.

[23] Y.Genin, P. Van Dooren, T.Kailath, J.Delsome and M.Morf, *On Σ-lossless transfer functions and related questions*, Linear Algebra Appl., **50** (1983) 251-275.

[24] G.Heinig G. and K.Rost, *Algebraic methods for Toeplitz-like matrices and operators*, OT13, Birkhäuser, Basel, 1984.

[25] H.Kimura, *Directional interpolation approach to H^∞-optimization and robust stabilization*, IEEE Trans. Automatic Control, **AC-32** (1987) 1085-1093.

[26] T.Kailath, S.Kung and M.Morf, *Displacement ranks of matrices and linear equations,* J. Math. Anal. and Appl., **68** (1979) 395-407.

[27] T.Kailath and V.Olshevsky, *Diagonal pivoting for partially reconstructible Cauchy-like Matrices, with applications to Toeplitz-like linear equations and to boundary rational matrix interpolation problems*, Linear Algebra and Its Applications, **254** (1997), 251-302.

[28] T.Kailath and V.Olshevsky, *Displacement structure approach to polynomial Vandermonde and related matrices*, *Linear Algebra and Its Applications*, **261** (1997) 49-90.

[29] I.V.Kovalishina and V.P.Potapov, *Indefinite metric in the Nevanlinna-Pick problem*, Amer. Math. Soc. Transl. **138(2)** (1988) 15-19. (Russian original 1974.)

[30] T.Kailath and A.H.Sayed, *Displacement structure : Theory and Applications*, SIAM Review, **37** (1995) 297-386.

[31] H.Lev-Ari and T.Kailath *State-space approach to factorization of lossless transfer functions and structured matrices*, Linear Algebra Appl., **162 - 164** (1992) 273 - 295.

[32] M.Morf, *Fast algorithms for multivariable systems*, Ph.D. thesis, Department of Electrical Engineering, Stanford University, 1974.

[33] M.Morf, *Doubling algorithms for Toeplitz and related equations,* Proc. IEEE Internat. Conf. on ASSP, IEEE Computer Society Press, (1980), 954-959.

[34] R.Nevanlinna, *Über beschränkte analytishe Functionen*, Anal. Acad. Sci. Fenn., **32**(7) (1929) 1-75.

[35] V.Olshevsky, *Pivoting for structured matrices with applications,* 1997, http://www.cs.gsu.edu/˜matvro

[36] G.Pick, *Über die Beschräkungen Analytisher Functionen, Welche Durch Vorgegebene Functionswerte Bewirkt Werden*, Math. Ann., **77** (1916), 7-23.

[37] Y.Potapov, *The multiplicative structure of J-contractive matrix functions*, Amer. Math.Translations, **15** (1960) 131-244.

[38] M.Rosenblum and J.Rovniak, *Hardy classes and operator theory*, Oxford University Press, 1986.

[39] L.Sakhnovich, *Factorization problems and operator identities*, Russian Mathematical Surveys, **41(1)** (1986) 1-64.

[40] A.Sayed,T.Kailath, H.Lev-Ari and T.Constantinescu, *Recursive solutions of rational interpolation problems via fast matrix factorization*, Integral Equations and Operator Theory, **20** (1994) 84-118.