Transactions of A. Razmadze
Mathematical Institute
Vol. **1??** (202?), ?–??

A FRAMEWORK FOR THE AUTOMATION OF WIENER–HOPF FACTORIZATION

NATALIIA ADUKOVA [1*], VICTOR ADUKOV[2] AND GENNADY MISHURIS[3]

Abstract. Despite the extensive development of the Wiener–Hopf factorization theory, serious challenges remain in its algorithmic implementation. In particular, there are no explicit formulas for constructing the factorization; the problem is not stable in the general case; the factorization is not unique; and there are no effective criteria for a canonical factorization. The aim of this work is to address these challenges by initiating a project devoted to the automation of the Wiener–Hopf factorization process for certain classes of matrix functions. We introduce the first version of an Automated Research System (ARS) Factorization, which consists of three packages: ExactMPF, ExactStableMPF, and CanonicalSNSF. The cornerstone of the ARS is the package ExactMPF. It simultaneously constructs the exact right and left factorizations for Laurent matrix polynomials in both stable and non-stable cases when such factorizations exist. The package supports parallel and hybrid symbolic–numerical computations. The package ExactStableMPF focuses only on the exact right factorization of Laurent matrix polynomials in the stable case, which is typically required in applications. By restricting to this setting, it achieves improved computational efficiency compared to the universal package ExactMPF. In addition, it implements P-normalization to ensure uniqueness of the stable factorization. The third package, CanonicalSNSF, is intended for the approximate construction of the canonical factorization with guaranteed accuracy for strictly non-singular 2×2 matrix functions. The theoretical results underlying the algorithms implemented in the packages are presented in this article.

1. Introduction

The Wiener–Hopf factorization problem occupies a central position in the theory of linear operators and singular integral equations and has a long history of development [20, 25, 26, 34–36, 39, 40]. At the same time, it plays a fundamental role in many applied problems arising in elasticity, diffraction theory, wave propagation and electromagnetics [1,21,33]. Despite this substantial theoretical progress, the matrix Wiener–Hopf factorization problem remains a challenging and unsolved problem [2, 37], particularly from the viewpoint of effective and computational applications. Among the representative contributions are the exact factorization algorithms developed in [8, 9], as well as a series of computational approaches proposed by L. Ephremidze and co-authors [22–24, 31].

In this paper we present the first stage of the development of an Automated Research System (ARS) Factorization in CAS Maple. The ARS is designed to automate research in the field of Wiener–Hopf factorization of matrix functions. The ultimate goal of the project is to develop a set of software products that would allow for an **arbitrary** invertible matrix function from the matrix Wiener algebra to determine whether it allows the stable Wiener–Hopf factorization and, in the case of a positive answer to this question, to construct it with guaranteed accuracy. The project aims to develop packages for constructing factorizations in such important classes as matrix polynomials, rational matrix functions, strictly non-singular 2×2 matrix functions, matrix functions from Khrapkov's class, and other related classes.

The cornerstone of the ARS is the package ExactMPF [8]. It automates the construction of both left and right Wiener–Hopf factorizations for Laurent matrix polynomials when this can be done with exact calculations. The advantage of such calculations is that in this case there are no problems with instability. Unfortunately, error-free calculations come at a high price: as the degree of the

---

2020 *Mathematics Subject Classification.* Primary 47A68, 30E25, 15-04 · Secondary 15A23.

Key words and phrases. Wiener–Hopf factorization; matrix polynomials; automation of calculations; error-free calculations.

*Corresponding author.

polynomial increases, the execution time of the task increases significantly. Despite this, the package was successfully applied to solving the discrete analogue of non-linear Schrödinger equation by the inverse scattering transform method [10], and to finding, with guaranteed accuracy, the canonical factorization of strictly non-singular matrix functions [7].

In applications, only the right or left factorization is often required, and besides in the stable case. For this reason, a simplified version of the package ExactMPF was developed. The new package ExactStableMPF was tested on the examples from [7] and demonstrated a significant decrease in the execution time. Moreover, the package performs normalization of the factorization which guarantees its uniqueness.

The third package included in the initial version of ARS Factorization is CanonicalSNSF. It allows one to find, with guaranteed accuracy, the canonical factorization of strictly non-singular 2×2 matrix functions. The algorithm for this package is based on a new approach to the canonical factorization of arbitrary matrix functions proposed in [7]. We also used the explicit formulas for the canonical factorization of strictly non-singular 2×2 Laurent matrix polynomials from [17].

The paper is organised as follows. In Sec. 2 we recall the basic concepts in the theory of the Wiener–Hopf factorization. In Sec. 3 we formulate the basic problems arising in this theory. The theoretical basis and design of the package ExactMPF are described in Sec. 4. These questions are discussed in detail in [8]. The ExactStableMPF package is described in Sec. 5. Sec. 6 is devoted to the CanonicalSNSF package.

For the convenience of the reader, we provide a brief summary of the notation used throughout the paper.

The variety of notation reflects the notation adopted in the principal references on which the present work is based. We retain this notation in order to maintain consistency with the original sources and to facilitate comparison with the cited results.

Table 1. Summary of notation used throughout the paper

Notation	**Meaning**
$A(t) = A_-^r(t)D^r(t)A_+^r(t)$	Right Wiener–Hopf factorization of a matrix function $A(t)$.
$A(t) = A_+^l(t)D^l(t)A_-^l(t)$	Left Wiener–Hopf factorization of a matrix function $A(t)$.
$A(t) = A_-(t)D(t)A_+(t)$	Right factorization; the superscript r is omitted throughout most of the paper.
$A(t) = G_-(t)D(t)G_+(t)$	Factorization obtained from a given factorization by the Gohberg–Krein theorem on the general form of factorization factors.
$A(t) = C_-(t)D(t)C_+(t)$	Canonically normalized right factorization.
$A(t) = D_b(t)B_-(t)B_+(t)$	Birkhoff factorization used in the normalization theory.
$a(t) = r_-(t)d_r(t)r_+(t)$	Explicitly constructed right factorization of a matrix polynomial.
$a(t) = l_+(t)d_l(t)l_-(t)$	Explicitly constructed left factorization of a matrix polynomial.

2. The Wiener–Hopf factorization. Basic concepts

In the theory of linear operators [20, 25, 36, 39] and in various branches of physics, mechanics and engineering [1, 2, 7, 8, 10, 17, 21, 32, 33, 37] the Riemann boundary value problem is an extremely efficient tool. In most applications of the Riemann problem with the matrix coefficient $A(t)$, a special system of solutions, namely a canonical system, is used. A matrix function formed from these solutions is called *a canonical matrix* of the problem. At present, the problem of finding the canonical matrix is formulated as the factorization problem for $A(t)$. A detailed exposition of the factorization theory can be found in [20, 25, 34].

Let Γ be a simple piecewise smooth closed contour in the complex plane $\mathbb{C}$ bounding the domain D_+. We assume that $0 \in D_+$. The complement of $D_+ \cup \Gamma$ in the closed complex plane, $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$, is denoted by D_-. Let $A(t)$ be a $p \times p$ matrix function on Γ. We assume that $A(t)$ is continuous and invertible on Γ.

The representation of a matrix function $A(t)$

$$A(t) = A^l_+(t) D^l(t) A^l_-(t), \ t \in \Gamma,$$

is called *the left Wiener–Hopf factorization* of the matrix function [20]. Here, $A^l_\pm(t)$ are continuous and invertible matrix functions on Γ that admit analytic continuation into $D_\pm$ and their continuations $A^l_\pm(z)$ are invertible in the respective domains; $D^l(t) = \operatorname{diag}[t^{\lambda_1}, \dots, t^{\lambda_p}]$, where integers $\lambda_1, \dots, \lambda_p$ are called *the left partial indices* of $A(t)$. It is always possible to assume that $\lambda_1 \geq \dots \geq \lambda_p$. Moreover, $\lambda_1 + \dots + \lambda_p = \varkappa$, where $\varkappa = \operatorname{ind} \det A(t)$ is the winding number of the determinant of the matrix $A(t)$. If $\lambda_1 = \dots = \lambda_p = 0$, then the factorization is called *left canonical*.

Similarly, *the right Wiener–Hopf factorization* of $A(t)$ is the representation

$$A(t) = A^r_-(t) D^r(t) A^r_+(t), \ t \in \Gamma,$$

where $A^r_\pm(t)$ have the same properties as $A^l_\pm(t)$, $D^r(t) = \operatorname{diag}[t^{\rho_1}, ..., t^{\rho_p}]$, and $\rho_1 \leq \dots \leq \rho_p$, $\rho_1 + \dots + \rho_p = \varkappa$, are called *the right partial indices* of $A(t)$. If $\rho_1 = \dots = \rho_p = 0$, then the factorization is called *right canonical*.

Even when both factorizations exist, the left and right indices are usually different sets of integers, and the construction of the left and right factorizations is usually considered as two separate problems.

These definitions require that $A^{l,r}_\pm(t)$ be continuous on Γ. This type of factorization is often called *standard* [20]. Also, the factorization is considered in some Banach algebras of continuous matrix functions on Γ. Typical examples of such algebras are the algebra $C^{p\times p}(\Gamma)$ of all matrix functions continuous on Γ, the algebra $H^{p\times p}_\mu(\Gamma)$, $0 < \mu < 1$, of matrix functions with Hölder entries, and the matrix Wiener algebra $W^{p\times p}$.

The existence of factorization for matrix functions was studied in the works of J. Plemelj, F.D. Gakhov, N.I. Muskhelishvili, N.P. Vekua, B.V. Hvedelidze, I.C. Gohberg, M.G. Krein. The existence of factorization in the algebra $H^{p\times p}_\mu(\Gamma)$ was proved in [35, 40], in the algebra $W^{p\times p}$ in [26].

In this article we will only deal with the standard factorization. Moreover, in Section 5 we will need an approximation of the matrix function $A(t)$ by matrix polynomials in t, t^{-1}. It cannot be done in the Banach algebra $H^{p\times p}_\mu(\Gamma)$ since it is non-separable. Thus we arrive at the factorization in the matrix Wiener algebra $W^{p\times p}$. This simplifies the implementation of theoretical results in a software product.

The definition of the factorization in $W^{p\times p}$ can be done without explicit use of complex analysis. Moreover, the existence of the factorization in $W^{p\times p}$ can only be proved using tools of functional analysis [17]. Sometimes this approach allows one to obtain the factorization in the explicit form.

The matrix Wiener algebra $W^{p\times p}$ consists of $p \times p$ matrix functions with entries from the Wiener algebra W. Thus any $A(t) \in W^{p\times p}$ expands into an absolutely convergent matrix Fourier series, $A(t) = \sum_{k=-\infty}^{\infty} A_k t^k$, $|t| = 1$, such that $\sum_{k=-\infty}^{\infty} \|A_k\| < \infty$. Here, A_k belong to the algebra $\mathbb{C}^{p\times p}$ complex matrices equipped with any matrix multiplicative norm on $\mathbb{C}^{p\times p}$, preserving unity.

The algebra $W^{p\times p}$ becomes a Banach algebra with a norm $\|A\|_W = \sum_{k=-\infty}^{\infty} \|A_k\|$. Typically we will use operator norms $\|\cdot\|_1$ or $\|\cdot\|_\infty$ on $\mathbb{C}^{p\times p}$. The corresponding norm in the matrix Wiener algebra is denoted as $\|\cdot\|_{W,1}$ or $\|\cdot\|_{W,\infty}$. Obviously, any matrix polynomial $\sum_{k=-N}^{N} A_k t^k$ in t, t^{-1} belongs to $W^{p\times p}$. We call such polynomial *the Laurent matrix polynomial*.

Let us define

$$W^{p\times p}_+ = \{A(t) \in W^{p\times p} : A(t) = \sum_{k=0}^{\infty} A_k t^k\}, W^{p\times p}_- = \{A(t) \in W^{p\times p} : A(t) = \sum_{k=-\infty}^{0} A_k t^k\}.$$

These are closed subalgebras of $W^{p\times p}$ with the unit I_p, I_p is the $p \times p$ unit matrix. The algebra $W^{p\times p}_+$ ($W^{p\times p}_-$) consists of matrix functions from $W^{p\times p}$ that admit analytic continuation into the domain $|z| < 1$ ($|z| > 1$). We denote the groups of invertible elements of $W^{p\times p}$, $W^{p\times p}_\pm$ by $GW^{p\times p}$, $GW^{p\times p}_\pm$, respectively.

Let us reformulate the factorization problem for algebra $W^{p\times p}$. Let $A(t)$ be a matrix function from $GW^{p\times p}$. *The left Wiener–Hopf factorization* of $A(t)$ is its representation in the following form:

$$A(t) = A^l_+(t)D^l(t)A^l_-(t),\ |t| = 1. \tag{2.1}$$

where $A^l_\pm(t) \in GW^{p\times p}_\pm$, $D^l(t) = \mathrm{diag}\,[t^{\lambda_1}, \dots, t^{\lambda_p}]$.

The right Wiener–Hopf factorization of $A(t)$ is defined analogously:

$$A(t) = A^r_-(t)D^r(t)A^r_+(t). \tag{2.2}$$

Here $A^r_\pm(t) \in GW^{p\times p}_\pm$, $D^r(t) = \mathrm{diag}\,[t^{\rho_1}, \dots, t^{\rho_p}]$.

3. The Wiener–Hopf factorization. Fundamental results and underlying obstacles

Let us point out the four main problems in the factorization theory for matrix functions. These problems represent the basic obstacles to the extensive use of the factorization.

3.1. **Obstacle I: Instability of the factorization.** We limit ourselves to considering only the right factorization

$$A(t) = A_-(t)D(t)A_+(t), \quad |t| = 1. \tag{3.1}$$

Definition 1. The partial indices $\rho_1, \dots, \rho_p$ of the matrix function $A(t)$ are called **stable** if for any small enough $\varepsilon > 0$ any matrix function $\widetilde{A}(t)$, satisfying the inequality $\|A(t) - \widetilde{A}(t)\|_W < \varepsilon$, possesses the same set of right partial indices as $A(t)$.

Definition 2. The factors $A_\pm(t)$ are called **stable** (or continuously depend on $A(t)$), if for any small enough $\varepsilon > 0$ there is $\delta > 0$ such that for any matrix function $\widetilde{A}(t)$, satisfying the inequality $\|A(t) - \widetilde{A}(t)\|_W < \delta$, among all possible factorizations of $\widetilde{A}(t)$ there exists the factorization $\widetilde{A}(t) = \widetilde{A}_-(t)\widetilde{D}(t)\widetilde{A}_+(t)$, for which $\|A_\pm(t) - \widetilde{A}_\pm(t)\|_W < \varepsilon$.

The clarification "among all possible factorizations" for $\widetilde{A}(t)$ is due to the fact that the factors $A_\pm(t)$ are not found in a unique way (see Obstacle III). Therefore, it is impossible to talk about the closeness of factors without choosing them in a special way, i.e. without normalizing the factorizations.

Definition 3. We will call the factorization **stable**, if the partial indices are stable under a small perturbation of $A(t)$ and factors $A_\pm(t)$ continuously depend on $A(t)$.

Recall some well-known facts on stability of the indices and the factors. There is a classical criterion of Gohberg–Krein–Bojarsky for stability of the partial indices [19, 26, 27]:

Theorem 3.1 (GKB-criterion)**.** *The right partial indices $\rho_1 \le \dots \le \rho_p$ are stable under a small perturbation of $A(t)$ if and only if $\rho_p - \rho_1 \leqslant 1$.*

Unfortunately, the GKB-criterion is not effective, since there are no explicit formulas for the partial indices (see Obstacle II).

There are two possible cases. In the first case all partial indices are equal, $\rho_1 = \dots = \rho_p$. Then $A(t) = t^{\rho_1}A_-(t)A_+(t)$, i.e. the factorization of $A(t)$ is equivalent to the canonical factorization of $t^{-\rho_1}A(t)$.

In the second case we have $\rho_p - \rho_1 = 1$. Therefore, $\rho_1 = \dots = \rho_{p-r} = \nu$, $\rho_{p-r+1} = \dots = \rho_p = \nu + 1$. Here the numbers ν and r are determined from the relation $\varkappa := \mathrm{ind}\det A(t) = \nu p + r$, where r is the remainder of dividing the index of determinant $\varkappa$ on the order p of matrix function. In this case, $D(t)$ has the following block form:

$$D(t) = \begin{pmatrix} t^\nu I_{p-r} & 0 \\ 0 & t^{\nu+1} I_r \end{pmatrix}, \tag{3.2}$$

where I_{p-r}, I_r are the unit matrices of corresponding order.

With regard to the stability of the factors, it is known that a necessary condition is the coincidence of the partial indices of the original and perturbed matrix functions [34, Theorem 6.14]. The main result in this direction is a theorem attributed to M.A. Shubin [38].

Theorem 3.2. *Let $A(t)$ admit the Wiener–Hopf factorization*

$$A(t) = A_-(t)D(t)A_+(t), \quad |t| = 1.$$

Then for any small enough $\varepsilon > 0$ there exists $\delta > 0$ such that for any matrix functions $\widetilde{A}(t)$ satisfying the inequality $\|A(t) - \widetilde{A}(t)\|_W < \delta$ and **having the same set of partial indices as** *$A(t)$ among all possible factorizations of $\widetilde{A}(t)$ there is a factorization $\widetilde{A}(t) = \widetilde{A}_-(t)D(t)\widetilde{A}_+(t)$, for which*

$$\|A_\pm(t) - \widetilde{A}_\pm(t)\|_W < \varepsilon.$$

Therefore, if the partial indices are stable then the factors will be also stable. The explicit criterion of the stability of the indices is known for matrix polynomials [5]. This criterion will be used in package ExactStableMPF.

Shubin's theorem is also non-constructive since in the general case the partial indices of $\widetilde{A}(t)$ are not known. Moreover, even under a priori assumptions about the coincidence of the partial indices of $A(t)$ and $\widetilde{A}(t)$, the theorem does not specify how to choose the factorizations for $A(t)$, $\widetilde{A}(t)$ for which the factors $A_\pm(t)$ and $\widetilde{A}_\pm(t)$ will be close to each other.

3.2. **Obstacle II: Absence of explicit factorization formulas.** Works on this topic (see, for example, [5, 7, 8, 10–12, 17, 28, 29, 32, 37]) contain the terms "explicit solution", "solutions in closed-form", "effective solution", and "constructive solution". Often these terms (or part of them) are considered as synonyms.

The conceptual difficulty is that it is impossible to give precise formal definitions of these terms. The interpretation of these concepts varies among authors, and no generally accepted terminology currently exists.

A first attempt to classify these notions was proposed in [8]. In the present paper we discuss this issue in greater detail.

By **an explicit solution (or a closed-form solution)** *to the factorization problem we will mean an expression (formula) that is formed using a finite number of operations from a predetermined list of basic operations (LBO). The basic operations are considered a priori as explicit. The list may vary depending on the context.*

If the factorization is built in several explicit steps, then it makes sense to consider this as a separate case.

By **a constructive (or effective) solution to the factorization problem** *we will mean an algorithmic procedure that constructs the factorization in a finite number of steps. Each step is implemented using a finite number of operations from a predetermined list of basic operations (LBO).* These steps we will call **admissible**.

To clarify the distinction between explicit and constructive (effective) solutions of the factorization problem, we formulate it more precisely:

- in an explicit solution, the algorithm terminates with formulas that directly determine the factorization. In a constructive (effective) solution, such formulas are not available.
- for an explicit solution, the number of steps required to construct the factorization can be specified *a priori*. For a constructive solution, the number of steps is determined during the execution of the algorithm.
- as a rule, the result of an explicit solution can be formulated in the form of a theorem, whereas a constructive solution is usually described only as an algorithmic procedure.

The difference between these two approaches can be illustrated by comparing the explicit factorization of meromorphic matrix functions obtained in [11] with the constructive computation of a canonical matrix for a rational matrix function presented in [40].

In the first case, explicit formulas for the partial indices (Theorem 0.1) and explicit formulas for the factorization factors (Theorem 4.1) are derived.

In the latter case, an algorithm is described (Chapter 1, Section 6) that reduces the factorization problem for a rational matrix function to the corresponding problem for a polynomial matrix. For the polynomial case, an algorithm for constructing a normal system of solutions is provided. The subsequent construction of a canonical system of solutions is based on the fact that, once a normal system is known, a canonical system can be obtained by a finite computational procedure.

Therefore, the latter approach should be regarded as a constructive (effective) solution of the factorization problem rather than an explicit one.

An additional restriction should be imposed on the notion of explicit (constructive) solution. For an explicit or constructive method to be practically useful, it should be computer-realizable.

This requirement is rather restrictive because of the instability of the factorization problem. If at least one operation from the LBO is performed approximately, the correctness of the final result can no longer be guaranteed. This difficulty can be avoided by using error-free calculations.

We will say that an explicitly (constructively) solvable factorization problem can be solved **exactly** *if the input numerical data belong to the Gaussian field $\mathbb{Q}(i)$ consisting of complex rational numbers, and all operations from the LBO can be realized in the rational arithmetic.*

The requirement of exact solvability significantly restricts the applicability of explicit and constructive methods. Below this requirement will be explicitly formulated for matrix polynomials. Exact calculations will be used in packages ExactMPF and ExactStableMPF.

Sometimes the second reason for the violation of the condition of computer-realizability arises. This condition is impossible to overcome, if it is necessary to work simultaneously with an infinite set of numerical data. Such operations cannot be implemented directly on a computer.

For example, the factorization of scalar functions has always been considered as a classical example of an explicitly solvable factorization problem. Recall a factorization procedure for $A(t) \in GW$.

Let $G(t) := \log(t^{-\varkappa}A(t)) = \sum_{k=-\infty}^{\infty} G_k t^k$, the projector $\mathscr{P}_+$ operates according to the rule: $\mathscr{P}_+ G(t) = \sum_{k=0}^{\infty} G_k t^k$, and $\mathscr{P}_-$ is the complementary projector. Then

$$A(t) = exp\,(\mathscr{P}_+ G)\, t^{\varkappa} exp\,(\mathscr{P}_- G) \tag{3.3}$$

is the Wiener–Hopf factorization. Let us define the following LBO: finding the index $\varkappa$ of $A(t)$, finding function log, finding the Fourier coefficients of G, the application of Sokhotsky-Plemelj formula, finding function exp. Then (3.3) is the explicit formula with respect to the given LBO.

However, any function $f(t) \in W$ is uniquely determined by the infinite set $\{f_k\}_{k=-\infty}^{\infty}$ of its Fourier coefficients. Any operations on functions are operations on an infinite set of numbers. Hence, if we take into account the computer-realizability, we cannot consider the formula (3.3) as explicit. This is precisely why we work with matrix polynomials, when we use explicit (constructive) methods. An extension of the class of matrix functions is possible if we will use approximate methods.

3.3. **Obstacle III: Non-uniqueness of the factors.** In order to obtain the unique factorization, we must normalize one of the factors $A_\pm(t)$. In presenting the theory of normalization we will follow the works [4, 15, 16].

There are the following motivations for the study of the normalization problem of the Wiener–Hopf factorization which guarantee the uniqueness of the factors $A_\pm(t)$.

- The need to compare factorizations of a given matrix function constructed by different methods.
- Refinement of Shubin's theorem regarding the choice of the factorization of the perturbed matrix function $\widetilde{A}(t)$.
- Obtaining effective estimates of absolute errors $\|A_\pm(t) - \widetilde{A}_\pm(t)\|_W$, when constructing an approximate factorization of $A(t)$.

The main tool in the normalization theory is the Gohberg–Krein theorem on the general form of the factorization factors $A_\pm(t)$ [25, Ch. VIII, Theorem 1.2]. Let us formulate it.

Let $\rho_1 \leq \ldots \leq \rho_p$ be the set of the right partial indices of $A(t) \in GW^{p\times p}$. We assume that this set contains s different numbers $\varkappa_1 < \ldots < \varkappa_s$ of multiplicity $k_1, \ldots, k_s$, respectively. Let $\mathscr{Q}_-(\rho_1, \ldots, \rho_p)$

be the set of all block triangular matrix functions of the form

$$Q_-(t) = \begin{pmatrix} Q_{11} & Q_{12} & \dots & Q_{1s} \\ 0 & Q_{22} & \dots & Q_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_{ss} \end{pmatrix}. \tag{3.4}$$

Here the block Q_{ij} has dimensions $k_i \times k_j$, the diagonal blocks Q_{ii} are constant invertible $k_i \times k_i$ matrices, and the off-diagonal blocks $Q_{ij}(t)$ are matrix polynomials in the variable t^{-1} of degree at most $\varkappa_j - \varkappa_i$. The set $\mathcal{Q}_-(\rho_1, \dots, \rho_p)$ is a subgroup of the group $GW_-^{p\times p}$.

Let $D(t) = \operatorname{diag}[t^{\rho_1}, \dots, t^{\rho_p}]$. Define a matrix function

$$Q_+(t) = D^{-1}(t)Q_-^{-1}(t)D(t).$$

$Q_+(t)$ has the same form as (3.4), only in this case $Q_{ij}(t)$ are matrix polynomials in t of degree at most $\varkappa_j - \varkappa_i$. Thus, $Q_+(z) \in GW_+^{p\times p}$.

The Gohberg–Krein theorem on the general form of factorization factors states that if (3.1) is a Wiener–Hopf factorization of the matrix function $A(t)$ with partial indices $\rho_1 \le \dots \le \rho_p$, then the representation

$$A(t) = G_-(t)D(t)G_+(t), \tag{3.5}$$

where $G_-(t) = A_-(t)Q_-(t)$, $G_+(t) = Q_+(t)A_+(t)$ is also a Wiener–Hopf factorization of $A(t)$ for any $Q_-(t) \in \mathcal{Q}_-(\rho_1, \dots, \rho_p)$. Moreover, any factorization $A(t)$ can be obtained from the original factorization (3.1) in a similar way with an appropriate choice of $Q_-(t)$.

Definition 4. The transition from the original factorization (3.1) to the factorization (3.5) by using any matrix function $Q_-(t) \in \mathcal{Q}_-(\rho_1, \dots, \rho_p)$ will be called **the normalization of the factorization (3.1) at infinity**. The matrix function $Q_-(t)$ is called the **normalization matrix**.

Our task is to produce in some sense a canonical choice of $Q_-(t)$. The main condition that determines the choice of a canonical representative $Q_-(t) \in \mathcal{Q}_-(\rho_1, \dots, \rho_p)$ will be related to the Birkhoff factorization of the matrix function $A(t)$. This factorization was introduced by G. Birkhoff [18] in connection with some problems for ordinary differential equations. The right Birkhoff factorization of $A(t)$ is its representation in the following form:

$$A(t) = D_b(t)B_-(t)B_+(t), \quad |t| = 1, \tag{3.6}$$

where $B_\pm(t) \in GW_\pm^{p\times p}(\mathbb{T})$ and $D_b(t) = \operatorname{diag}\left[t^{\beta_1}, \dots, t^{\beta_p}\right]$, here $\beta_1, \dots, \beta_p$ are the right Birkhoff indices of $A(t)$. In contrast to partial indices, the Birkhoff indices are not uniquely determined by the matrix function $A(t)$. However, among all possible sets of the Birkhoff indices there exists a set obtained by some permutation of the right partial indices. Thus, one of the Birkhoff factorizations can be written as

$$A(t) = PD(t)P^{-1}B_-(t)B_+(t), \quad |t| = 1, \tag{3.7}$$

where P is some permutation matrix.

The following definition of the normalization was proposed in the report of N. Adukova and V. Adukov on 13th ISAAC congress [6].

Definition 5. Let P be a permutation matrix of order p. The Wiener–Hopf factorization of the matrix function $A(t)$:

$$A(t) = C_-(t)D(t)C_+(t) \tag{3.8}$$

is called P-**normalized** if the following conditions are fulfilled:

(1) The matrix function $B_-(t) = PD^{-1}(t)P^{-1}C_-(t)D(t)$ belongs to the algebra $W_-^{p\times p}(\mathbb{T})$,
(2) $B_-(\infty) = P$.

We will consider P-**normalization** as canonical normalization of the factorization.

It turns out that if the P-normalized factorization $A(t)$ exists, then it is unique and generates a Birkhoff factorization $A(t)$.

Theorem 3.3 ([15], Theorem 1)**.** *Let the matrix function $A(t) \in GW^{p\times p}(\mathbb{T})$ admits a P-normalized Wiener–Hopf factorization:*

$$A(t) = C_-(t)D(t)C_+(t).$$

Then

1. *this Wiener–Hopf factorization generates a Birkhoff factorization by the formula*
$$A(t) = PD(t)P^{-1}B_-(t)B_+(t),$$
where $B_-(t) = PD^{-1}(t)P^{-1}C_-(t)D(t)$, $B_+(t) = C_+(t)$.
2. *the given P-normalized Wiener–Hopf factorization is unique.*

At present, the existence of P-normalization is proved for 2×2 matrix functions [15], for matrix functions with the stable partial indices [4], and for matrix functions with distinct partial indices [16]. It turns out that the existence of the P-normalization is equivalent to the existence of the so-called PLU-factorization or block PLU-factorization of the invertible numerical matrix $A_-(\infty)$. Here $A(t) = A_-(t)D(t)A_+(t)$ is the original factorization of $A(t)$.

In this work we will only need the normalization of the stable factorization which uses a block PLU-factorization of $A_0 := A_-(\infty)$.

Let us partition the matrix $A_0 = \begin{pmatrix} A^0_{11} & A^0_{12} \\ A^0_{21} & A^0_{22} \end{pmatrix}$ into the same blocks as $Q_-(t)$. If the block A^0_{11} is invertible, then we can represent A_0 as

$$A_0 = L_0U_0 = \begin{pmatrix} I_{p-r} & 0 \\ A^0_{21}(A^0_{11})^{-1} & I_r \end{pmatrix} \begin{pmatrix} A^0_{11} & A^0_{12} \\ 0 & D^0_{22} \end{pmatrix},$$

where $D^0_{22} = A^0_{22} - A^0_{21}(A^0_1)^{-1}A^0_{12}$. This is **block LU-factorization** of A_0.

If A^0_{11} is a singular matrix, then there exists a permutation matrix P such that $P^{-1}A_0$ admits the block LU-factorization. Thus A_0 admits the decomposition $A_0 = PL_0U_0$ which is called **a block PLU-factorization** of A_0. The permutation P is not uniquely found. Note that if there exists a Wiener–Hopf factorization $A(t) = A_-(t)D(t)A_+(t)$ for which $A_-(\infty)$ admits the block PLU–factorization with the given P, then it follows from the theorem on the general form of factorization that any Wiener–Hopf factorization $A(t)$ has this property.

Theorem 3.4 ([4], Proposition 1, Theorem 1)**.** *A matrix function $A(t)$ with the stable partial indices admits the P-normalized Wiener–Hopf*

$$A(t) = C_-(t)D(t)C_+(t)$$

if and only if for any Wiener–Hopf factorization

$$A(t) = A_-(t)D(t)A_+(t)$$

the matrix $A_-(\infty)$ admits the block PLU–factorization.

In this case the matrix $C_-(t)$ has the form

$$C_-(t) = P \begin{pmatrix} I_{p-r} + t^{-1}C^-_{11}(t) & t^{-2}C^-_{12}(t) \\ C^-_{21}(t) & I_r + t^{-1}C^-_{22}(t) \end{pmatrix} \tag{3.9}$$

.

The algorithm for obtaining a normalized factorization if we know some stable Wiener–Hopf factorization is presented in [4, Theorem 1]. This algorithm will be used in package ExactStableMPF.

3.4. **Obstacle IV: Absence of a stability criterion in the general case.** If $A(t)$ is not a Laurent matrix polynomial admitting the exact factorization, then the task of finding an approximate factorization arises. Due to the instability of the problem, we can only talk about finding a stable factorization. In [7], we proposed a new approach to solving this problem.

A basic idea of the new approach is as follows. Let $A(t)$ be an invertible matrix function from the Wiener algebra $W^{p\times p}$ and $\mathscr{E}\mathscr{F}^{p\times p} \subset W^{p\times p}$ be some class of matrix functions admitting the explicit (or effective) solution of the factorization problem. For our approach to be applicable, two conditions must be met.

(1) The matrix function $A(t)$ can be approximated by some $A_\delta(t) \in \mathscr{E}F^{p\times p}$ with a given accuracy δ: $\|A - A_\delta\|_W \leq \delta$.
(2) $A_\delta(t)$ admits the stable factorization $A_\delta(t) = A_\delta^-(t)D_\delta(t)A_\delta^+(t)$.

By definition of the stable factorization there is a neighbourhood $U(A_\delta, r_{sf}^\delta)$ of $A_\delta(t)$ of some radius r_{sf}^δ such that any matrix function that belongs to this neighbourhood has the same set of the right partial indices as $A_\delta(t)$.

If the condition

$$\delta < r_{sf}^\delta, \tag{3.10}$$

is fulfilled, then the original matrix function $A(t)$ belongs to $U(A_\delta, r_{sf}^\delta)$ and has the stable partial indices. Fig. 1 illustrates the criterion (3.10) geometrically.

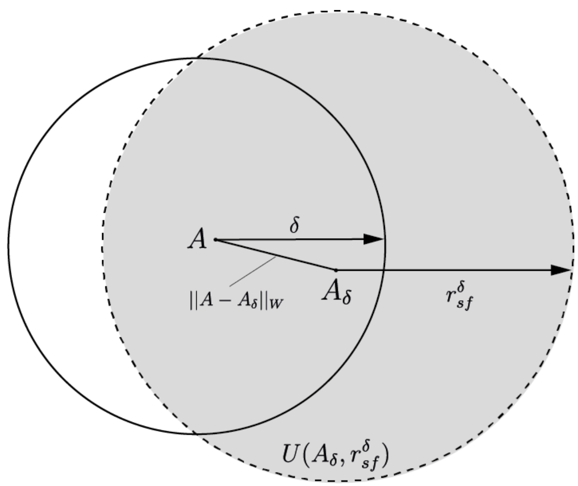

FIGURE 1. Neighbourhoods of A and A_δ for which $A(t)$ is expected to admit the stable factorization.

Based on the factorization $A_\delta(t) = A_\delta^-(t)D_\delta(t)A_\delta^+(t)$, effective estimates of r_{sf}^δ in terms of the factorization of $A_\delta(t)$ can be obtained. This estimate can be extracted from the proof of Theorem 4.1 [7]. Now the criterion (3.10) can be formulated as follows.

Theorem 3.5. *Let $A(t) \in W^{p\times p}$ such that*

(1) *There exists a matrix function $A_\delta(t) \in GW^{p\times p}$ admitting a stable factorisation*
$$A_\delta(t) = A_\delta^-(t)D_\delta(t)A_\delta^+(t)$$
and satisfying the inequality $\|A - A_\delta\|_W \leq \delta$ for some δ.
(2) *The inequality $q_\delta := \sigma \cdot \delta \cdot i(A_\delta) < 1$ is fulfilled. Here*
$$\sigma = \begin{cases} 1, & \text{for the stable factorization of I-type;} \\ 2, & \text{for the stable factorization of II-type} \end{cases},$$
and $i(A_\delta) = \|(A_\delta^+)^{-1}\|_W \|(A_\delta^-)^{-1}\|_W$.

Then $A(t)$ is an invertible matrix function admitting the stable factorization of the same type as $A_\delta(t)$.

Effective estimates of $\|A_\pm(t) - A_\delta^\pm(t)\|$ can be obtained in the same way as in the work [7, Corollary 5.1].

This new approach will be used in package CanonicalSNSF for the factorization of a strictly non-singular matrix function

$$A(t) = \begin{pmatrix} 1 & \beta(t) \\ \alpha(t) & 1 + \alpha(t)\beta(t) \end{pmatrix}, \quad \alpha(t), \beta(t) \in W.$$

In the next sections we describe in detail the packages ExactMPF, ExactStableMPF, and CanonicalSNSF. Since the paper is primarily intended as a practical guide for potential users of these packages, we include the minimum theoretical material necessary for understanding the underlying algorithms and their implementation. The theoretical results summarized here were obtained in the following earlier publications: [3–17].

4. Package ExactMPF

Our immediate objective is to develop a package for finding the Wiener–Hopf factorization of Laurent matrix polynomials. We have theoretical results for solving this problem [8, 9, 13, 14], but we must ensure the computer implementation of these results.

Due to the instability of the factorization problem, the last condition leads to a strong restriction of the class of factorizable matrix polynomials. For this reason, we will first consider the case of matrix polynomials for which the factorization can be found using error-free calculations. So we come to the problem of the exact factorization of Laurent matrix polynomials.

The package ExactMPF is designed to find the exact Wiener–Hopf factorization for Laurent matrix polynomials. The factorization of a Laurent matrix polynomial can be easily reduced to the factorization of a matrix polynomial.

4.1. An explicit factorization of a matrix polynomial.

Let $a(t)$ be a matrix polynomial, $\det a(t) = \Delta_-(t)t^{\varkappa}\Delta_+(t)$, $\Delta_-(\infty) = 1$, be a Wiener–Hopf factorization of the scalar polynomial $\det a(t)$.

Let $c_{-\varkappa}^{\varkappa} := \{c_{-\varkappa}, \dots, c_0, \dots, c_{\varkappa}\}$ be the sequence of complex $p \times p$ matrices that are the Laurent coefficients of $\Delta_-^{-1}(t)a(t)$. Following [13, 14] define for the sequence $c_{-\varkappa}^{\varkappa}$ indices $\mu_1, \dots, \mu_{2p}$ and essential factorization polynomials $R_1(t), \dots, R_{2p}(t); L_1(t), \dots, L_{2p}(t)$.

In terms of these concepts the left $a(t) = l_+(t)d_l(t)l_-(t)$ and right $a(t) = r_-(t)d_r(t)r_+(t)$ factorizations of $a(t)$ are constructed by the explicit formulas ([14], Theorem 3.2). Below we present these formulas.

Introduce the $p \times p$ matrix functions

$$\mathscr{R}_1(t) = \begin{pmatrix} R_1(t) & \dots & R_p(t) \end{pmatrix}, \quad \mathscr{L}_2(t) = \begin{pmatrix} L_{p+1}(t) \\ \vdots \\ L_{2p}(t) \end{pmatrix},$$

and

$$d_l(t) = \begin{pmatrix} t^{-\mu_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & t^{-\mu_p} \end{pmatrix}, \quad d_r(t) = \begin{pmatrix} t^{\mu_{p+1}} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & t^{\mu_{2p}} \end{pmatrix}.$$

Then the left $\lambda_1 \geq \dots \geq \lambda_p$ and right $\rho_1 \leq \dots \leq \rho_p$ partial indices and the factors $l_\pm(t)$, $r_\pm(t)$ of the respective factorizations of the matrix polynomial $a(t)$ are defined by the formulas

$$\lambda_1 = -\mu_1, \dots, \lambda_p = -\mu_p, \quad \rho_1 = \mu_{p+1}, \dots, \rho_p = \mu_{2p}, \tag{4.1}$$

$$l_-(t) = t^{\varkappa+1}\Delta_-(t)d_l^{-1}(t)\mathscr{R}_1^{-1}(t), \quad l_+(t) = t^{-\varkappa-1}\Delta_-^{-1}(t)a(t)\mathscr{R}_1(t), \tag{4.2}$$

$$r_-(t) = \Delta_-(t)\mathscr{L}_2^{-1}(t), \quad r_+(t) = \Delta_-^{-1}(t)d_R^{-1}(t)\mathscr{L}_2(t)a(t). \tag{4.3}$$

In these formulas we use the following basic operations: constructing the Wiener–Hopf factorization of the scalar polynomial $\det a(t)$, the evaluation of a finite set of Laurent coefficients of $\Delta_-^{-1}(t)a(t)$ by recurrence relations, finding the ranks and the null spaces for a finite set of block Toeplitz matrices $\{T_k\}_{k=-\varkappa}^{\varkappa}$. These operations can be realized by tools of the package LinearAlgebra in the computer algebra system (CAS) Maple. For this reason, we can consider the given factorization formulas to be **explicit (or effective)**. The successive steps of the factorization algorithm are described in detail in [8].

However, when instability of the factorization problem is taken into account, additional difficulties arise. In general case computer implementation of the basic operation of constructing the factorization of $\det a(t)$ requires approximate calculations. The operations of finding the ranks and the null spaces are unstable and now cannot be carried out correctly. Thus the given explicit factorization formulas are not computer-implementable.

One way to avoid instability issues is to use rational inputs and require that all basic operations are performed in rational arithmetic. Recall that in this case we are dealing with exact factorization.

Theorem 4.1 ([8, 9]). *Let $a(z)$ be a matrix polynomial over the field of Gaussian rationals $\mathbb{Q}(i)$. Then $a(z)$ admits the exact Wiener–Hopf factorization if and only if the polynomial $\det a(z)$ is exactly factorizable over $\mathbb{Q}(i)$.*

The condition that $\det a(t)$ admits the exact Wiener–Hopf factorization can be explicitly verified with the help of I. Schur test (see [8]).

4.2. **A description of the package ExactMPF.**. The exact factorization algorithm for $p \times p$ Laurent matrix polynomial $a(z) = \sum_{k=-M}^{N} a_k z^k$ has been implemented in the computer algebra system Maple as the package ExactMPF. The procedures included in the package are described in detail in [8].

The package works in the version Maple 11 or higher To access ExactMPF use the commands

```
> read("ExactMPF.txt");
> with(ExactMPF);
> with(LinearAlgebra);
```

To obtain the factorizations of $a(z)$ we run the procedure SolverExactMPF with the argument $a(z)$:

```
> lplus, dl, lminus, rminus, dr, rplus := SolverExactMPF(a):
```

The procedure SolverExactMPF returns the factors lplus, dl, lminus for the left factorization and the factors rminus, dr, rplus for the right factorization.

The user can view the package as a *black box*. It does not require prior knowledge of the algorithm by which the package operates, and the preconditions that the matrix function $a(z)$ must satisfy. The package processes any input data $a(z)$ and either returns a verified exact factorization of $a(z)$ or reports the reason why the (exact) factorization could not be constructed.

The package performs the following input validations:

(1) the input matrix $a(z)$ is a matrix function of a variable z;
(2) the input matrix $a(z)$ is a square matrix;
(3) the elements of the input $a(z)$ are polynomials in z, z^{-1};
(4) these polynomials are polynomials over the field $\mathbb{Q}(i)$;
(5) $\det a(z)$ does not vanish identically;
(6) $\det a(z)$ does not vanish on the unit circle $\mathbb{T}$;
(7) $\det a(z)$ admits the exact Wiener–Hopf factorization.

If the validation fails, the solver reports that the (exact) factorization is not possible and terminates.

From a technical point of view, it is convenient to denote a variable in the input matrix function by a fixed symbol. This explains test 1).

Test 7) is carried out as follows. First, $\det a(z)$ is factorized into irreducible factors over $\mathbb{Q}(i)$. Then each factor is checked using the Schur test to ensure that all its zeros lie inside or outside the unit circle. The result of the test may be negative for two reasons:

- $\det a(t)$ vanishes on the unit circle $\mathbb{T}$. In this case the Wiener–Hopf factorization of $a(t)$ does not exist.
- $\det a(z)$ does not vanish on the unit circle, but the exact factorization of $\det a(z)$ does not exist. Now the Wiener–Hopf factorization of $a(t)$ exists but it cannot be found by the exact methods.

Unfortunately, it is impossible to separate these two cases using symbolic methods. For this reason we verify condition 6) by numerical calculations. We find approximate zeros ζ_j^* of $\det a(z)$ with 20 significant digits, and verify that $||\zeta_j^*| - 1| > 10^{-20}$. The condition 6) is the only precondition that is recommended to be additionally checked by the user if the user wants to establish the reason why the package cannot build the exact factorization of $a(t)$.

In this and subsequent sections computations were performed on the laptop MSI with 8-core (16 threads) CPU AMD Ryzen 7 4800H, 16Gb, operating system Windows 10 Home.

Example. Consider

$$a := \begin{pmatrix} -4z^2+4z-\frac{2}{3} & 2z \\ -2z^2+2z-\frac{1}{3} & z^2+3z+3 \end{pmatrix}.$$

```
> lplus,dl,lminus,rminus,dr,rplus:=SolverExactMPF(a):
```

SolverExactMPF returns the following expressions for the factors of a(z):

```
> lplus,dl,lminus;
```

$$\begin{bmatrix} -4 & 2z \\ -2 & z^2+3z+3 \end{bmatrix}, \begin{bmatrix} z^2 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1-\frac{1}{z}+\frac{1}{6z^2} & 0 \\ 0 & 1 \end{bmatrix}$$

```
> rminus,dr,rplus;
```

$$\begin{bmatrix} -\frac{23}{3}-\frac{2}{9z} & -\frac{28}{3}-\frac{5}{18z} \\ -\frac{121}{9}-\frac{475}{36z} & -\frac{587}{36}-\frac{581}{36z} \end{bmatrix}, \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix}, \begin{bmatrix} -\frac{2304}{17}-\frac{1676z}{17} & -\frac{180}{17}-\frac{336z}{17} \\ \frac{1884}{17}+\frac{1384z}{17} & \frac{144}{17}+\frac{276z}{17} \end{bmatrix}$$

The execution time was 0.094 seconds.

Even for this simple example, constructing the factorization manually would require several hours of computations. All other examples in the given work cannot be solved without the use of a computer. This emphasizes the importance of the requirement of the computer-realizability for a factorization algorithm.

4.3. **Acceleration of SolverExactMPF..** Error-free calculations allow to avoid troubles with instability of the factorization problem. However, this often comes at the cost of a significant increase in the computation time. This version of the package provides two ways to reduce the execution time. The first way is to apply parallel computing if the hardware allows it. For this, the solver ParSolverExactMPF is used instead of SolverExactMPF. In this case ranks and null spaces of matrices from the set $\{T_k\}_{k=-\varkappa}^{\varkappa}$ are found by parallel computations.

The second way is to use the option "symnum" that is built into the procedure SolverExactMPF. It can be evaluated by the command:

```
> lplus,dl,lminus,rminus,dr,rplus:=SolverExactMPF(a,"symnum"):
```

In this mode, SolverExactMPF finds the ranks of the matrices T_k uses the procedure NumRank included in the package ExactMPF instead of the command Rank from the package LinearAlgebra.

NumRank finds a numerical rank of a $N \times M$ matrix T with floating-point entries. A commonly used approach is to employ the numerical rank, which is defined using the singular value decomposition method (SVD) [30].

It is known that perturbations of the singular values are of the same order of magnitude as the perturbations of the matrix entries. Thus, computation of the singular values is a stable procedure and this property has led to the definition of a numerical τ-rank (see [30]). A numerical τ-rank of the matrix T is the number of singular values greater than some tolerance τ, i.e. the singular values that are less than or equal to τ are considered as perturbations of the zero singular values.

Thus, the tolerance τ specifies a way of separating the non-zero singular numbers from those assumed to be zero before the perturbation. Hence, the notion of τ-rank is useful when there exists a well-defined gap between non-zero and zero singular values of the matrix. Even for a matrix of the full rank, this requirement may not be fulfilled (if the matrix is ill-conditioned with the respect to the spectral condition number). The choice of the parameter τ may be highly sensitive and therefore represents a significant practical difficulty. In the procedure NumRank, a default value of the tolerance is $\tau(T) = \max\{N, M\} \cdot \sigma_1(T) \cdot 10^{-\text{Digits}}$, where $\sigma_1(T)$ is the greatest singular value of the matrix T.

Further, τ-ranks of the block Toeplitz matrices T_k are used to define τ-indices of the sequence $c_{-\varkappa}^{\varkappa}$ instead of the exact indices of this sequence. All remaining calculations are performed in the exact arithmetic. If the output validations are successful, the replacement of the indices by τ-indices may be considered as justified. Using this mode, one can significantly speed up the calculations.

The procedure (Par)SolverExactMPF performs the following output tests for the constructed factorization.

(1) It is verified that $\sum_{k=1}^{p}\mu_k = -\varkappa$, $\sum_{k=p+1}^{2p}\mu_k = \varkappa$.
(2) It is verified that the factors $l_+(z), r_+(z)$ $(l_-(z), r_-(z))$ are matrix polynomials in z (in z^{-1})
(3) It is verified that $(\Delta_\pm(z))^{-1}\det l_\pm(z)$, $(\Delta_\pm(z))^{-1}\det r_\pm(z)$ are constants.

If the test results are positive, it means that the factorizations have been constructed correctly.

Example. Let $p_1 := -2z^2+2z-\frac{1}{3}$, $p_2 := z^2+2z+3$. Consider the 5×5 matrix polynomial

$$b := \begin{pmatrix} z^3p_1^2+z^2+z+1 & z^2p_1+iz^3+1 & z+1 & iz^6+1 & p_2 \\ z^7+z^4+z & zp_1+z^4 & z^6+z^5 & z^6 & z^6p_2 \\ 2z & z+i & z^2+1 & iz^3+1 & p_2 \\ z(z-1) & z & z-1 & z^4+z-1 & p_1 \\ zp_1 & 0 & p_1 & p_1 & p_1p_2 \end{pmatrix}.$$

SolverExactMPF builds the factorizations of $b(z)$ in 75.734 sec.
Let us apply ParSolverExactMPF with the parallel computations.

```
> lplus,dl,lminus,rminus,dr,rplus:=ParSolverExactMPF(b):
```

"Version from 15.06.2025"
"Parallel Computation on 16 processors"
...
"The exact factorization is constructed"
"The total computation time is 22.875 seconds"

```
> dl,dr;
```

$$\begin{bmatrix} z^5 & 0 & 0 & 0 & 0 \\ 0 & z^4 & 0 & 0 & 0 \\ 0 & 0 & z^3 & 0 & 0 \\ 0 & 0 & 0 & z^3 & 0 \\ 0 & 0 & 0 & 0 & z^3 \end{bmatrix}, \quad \begin{bmatrix} z^3 & 0 & 0 & 0 & 0 \\ 0 & z^3 & 0 & 0 & 0 \\ 0 & 0 & z^4 & 0 & 0 \\ 0 & 0 & 0 & z^4 & 0 \\ 0 & 0 & 0 & 0 & z^4 \end{bmatrix},$$

Using the mode "symnum" gives the following results:

```
> lplus,dl,lminus,rminus,dr,rplus:=SolverExactMPF(b,"symnum"):
```

"Version from 15.06.2025"
"Computation on one processor"
...
"The method: symbolic-numeric"
...
"The exact factorization is constructed"
"The total computation time is 20.547 seconds"

Here we show only dl, dr, since the expressions for the factors are rather cumbersome. We see that both acceleration methods give approximately the same results.

5. Package ExactStableMPF

ExactMPF is a universal package: it simultaneously constructs the exact right and left factorizations in both stable and non-stable cases. Due to this universality, the execution time for a Laurent matrix polynomial of degree N increases significantly as N grows. The acceleration provided by the parallel or symbolic-numerical options is often insufficient. Meanwhile the information that the package ExactMPF delivers is redundant for many users. For example, in many applications, only the right (or left) factorization is required. If a user focuses on constructing an approximate factorization, then only the stable case should be considered.

To find an approximate solution with guaranteed accuracy, the factorization must be normalized to obtain a unique solution. The problem of finding the normalized right exact stable factorization of a Laurent matrix polynomial is solved by the package ExactStableMPF. The theoretical results on criteria for stable factorisation and its normalization can be found in [4,5,15,16]. The implementation of these theoretical results in this case has not yet been carried out.

Below, we present the theoretical results needed for constructing the normalized stable factorization. By virtue of the GKB-criterion the right partial indices $\rho_1 \le \ldots \le \rho_p$ are stable if and only if $\rho_p - \rho_1 \le 1$. We distinguish between the two cases of stability. Let us define the numbers ν, r from the relation $\varkappa = \nu p + r$, $0 \le r < p$. **The first case of stability** is the case, when $\rho_p - \rho_1 = 0$, i.e. $r = 0$. Now the factorization has the form

$$a(t) = t^{\nu} r_-(t) r_+(t), \ \ t \in \mathbb{T}. \tag{5.1}$$

and we can easily normalize it by the condition $r_-(\infty) = I_p$. **The second case of stability** occurs when $\rho_p - \rho_1 = 1$, i.e. $\rho_1 = \cdots = \rho_{p-r} = \nu, \ \rho_{p-r+1} = \cdots = \rho_p = \nu + 1$.

5.1. **An algorithm for the stable I factorization.** First, we will specify the stability criterion and the algorithm for constructing factorization in the first case of stability. Let us introduce the matrices

$$T_\nu = \begin{pmatrix} c_\nu & c_{\nu-1} & \cdots & c_{-\varkappa} \\ c_{\nu+1} & c_\nu & \cdots & c_{-\varkappa+1} \\ \cdots & \cdots & \cdots & \cdots \\ c_\varkappa & c_{\varkappa-1} & \cdots & c_{-\nu} \end{pmatrix}, \ \overline{T}_{\nu-1} = \begin{pmatrix} c_{\nu-1} & c_{\nu-2} & \cdots & c_{-\varkappa} \\ c_\nu & c_{\nu-1} & \cdots & c_{-\varkappa+1} \\ \cdots & \cdots & \cdots & \cdots \\ c_{\varkappa-1} & c_{\varkappa-2} & \cdots & c_{-\nu} \end{pmatrix}. \tag{5.2}$$

The matrix polynomial $a(t)$ admits the stable factorization of I-type if and only if the matrix T_ν has full rank (see [5]). If this condition is fulfilled then $\dim \ker_L \overline{T}_{\nu-1} = p$. Here $\ker_L \overline{T}_{\nu-1}$ is the left kernel of this matrix.

Let $\det a(z) = \Delta_-(z) z^{\varkappa} \Delta_+(z)$ be the Wiener–Hopf factorization of $\det a(z)$ and $L_{p+1}, \ldots, L_{2p}$ any basis of $\ker_L \overline{T}_{\nu-1}$. Divide the vector L_j, $p+1 \le j \le 2p$, into the blocks $L_k^j \in \mathbb{C}^{1\times p}$: $L_j = \begin{pmatrix} L_0^j & L_1^j & \cdots & L_{\varkappa-\nu}^j \end{pmatrix}$, and define the generating row polynomials $L_j(z) = L_0^j + L_1^j z^{-1} + \ldots + L_{\varkappa-\nu}^j z^{-\varkappa+\nu}$.

Then $\mathscr{L}_2(z) := \begin{pmatrix} L_{p+1}(z) \\ \vdots \\ L_{2p}(z) \end{pmatrix}$ is a $p \times p$ matrix polynomial in z^{-1}, $\det \mathscr{L}_2(z)$ does not vanish on the unit circle $\mathbb{T}$, and the matrix $\mathscr{L}_2(\infty) := \begin{pmatrix} L_{p+1}(\infty) \\ \vdots \\ L_{2p}(\infty) \end{pmatrix}$ is invertible. Denote $\mathfrak{L}(z) = \mathscr{L}_2^{-1}(\infty)\mathscr{L}_2(z)$.

Then the normalized factors of the right factorization (5.1) are constructed by the formulas:

$$r_-(t) = \Delta_-(t)\mathfrak{L}^{-1}(t), \quad r_+(t) = t^{-\nu}\Delta_-^{-1}(t)\mathfrak{L}(t)a(t). \tag{5.3}$$

If the entries of $a(z)$ are polynomials over $\mathbb{Q}(i)$ and $\det a(z)$ admits an exact factorization, then this factorization of $a(z)$ is exact.

5.2. **An algorithm for the stable II factorization.** Now we consider the second case of stability. Let us introduce the matrices

$$T_{\nu+1} = \begin{pmatrix} c_{\nu+1} & c_\nu & \cdots & c_{-\varkappa} \\ c_{\nu+2} & c_{\nu+1} & \cdots & c_{-\varkappa+1} \\ \cdots & \cdots & \cdots & \cdots \\ c_\varkappa & c_{\varkappa-1} & \cdots & c_{-\nu-1} \end{pmatrix}, \ \overline{T}_{\nu} = \begin{pmatrix} c_{\nu} & c_{\nu-1} & \cdots & c_{-\varkappa} \\ c_{\nu+1} & c_\nu & \cdots & c_{-\varkappa+1} \\ \cdots & \cdots & \cdots & \cdots \\ c_{\varkappa-1} & c_{\varkappa-2} & \cdots & c_{-\nu-1} \end{pmatrix}. \tag{5.4}$$

If $\nu > 0$ then $a(z)$ admits the stable factorization of II-type if and only if

$$\operatorname{rank} T_{\nu+1} = (\varkappa - \nu)p, \quad \operatorname{rank} T_\nu = (\varkappa+1)p - \varkappa. \tag{5.5}$$

In the case of $\nu = 0$ the stability criterion is

$$\operatorname{rank} T_1 = \varkappa p. \tag{5.6}$$

To construct the factorization for the second case of stability, we need the matrices $\overline{T}_\nu$ and $\overline{T}_{\nu-1}$. The spaces of generating polynomials of vectors from $\ker_L \overline{T}_k$ we denote $\overline{\mathscr{N}}_k^L$.

Let $\varkappa = \nu p + r$, $\nu \geq 0$, $r > 0$, and $\rho_1 = \ldots = \rho_{p-r} = \nu$, $\rho_{p-r+1} = \ldots = \rho_p = \nu + 1$.

Then $\dim \overline{\mathscr{N}}_{\nu-1}^L = p+r$, $\dim \overline{\mathscr{N}}_\nu^L = r$, and $\overline{\mathscr{N}}_{\nu-1}^L = (\overline{\mathscr{N}}_\nu^L \oplus z^{-1}\overline{\mathscr{N}}_\nu^L) \oplus \overline{\mathscr{H}}_{\nu-1}^L$, where $\dim \overline{\mathscr{H}}_{\nu-1}^L = p - r$.

Let $L_{2p-r+1}(z), \ldots, L_{2p}(z)$ be a basis of $\overline{\mathscr{N}}_\nu^L$ and $L_{p+1}(z), \ldots, L_{2p-r}(z)$ a basis of $\overline{\mathscr{H}}_{\nu-1}^L$.

Then the right factorization $a(z) = r_-(z)d_r(z)r_+(z)$ is constructed by the formulas

$$r_-(z) = \Delta_-(z)\mathscr{L}_2^{-1}(z),\ d_r(z) = \left(\begin{array}{c|c} z^\nu I_{p-r} & 0 \\ \hline 0 & z^{\nu+1} I_r \end{array}\right), r_+(z) = \Delta_-^{-1}(z) d_R^{-1}(z)\mathscr{L}_2(z)a(z). \tag{5.7}$$

Here $\mathscr{L}_2(z) = \begin{pmatrix} L_{p+1}(z) \\ \vdots \\ L_{2p}(z) \end{pmatrix}$.

Now we must normalize the constructed factorization. The algorithm for obtaining a P-normalized factorization from a given Wiener–Hopf factorization is presented in [4, Theorem 1]. The notion of a P-normalized factorization was considered in Sec. 2, Obstacle III.

5.3. **A description of the package** ExactStableMPF**.** The P-normalized exact stable factorization algorithm for a $p \times p$ Laurent matrix polynomial has been implemented in the computer algebra system Maple as the package ExactStableMPF.

To access ExactStableMPF use the commands

```
> read("ExactStableMPF.txt");
> with(ExactStableMPF);
> with(LinearAlgebra);
```

To obtain the factorizations of $a(z)$ we run SolverExactStableMPF with the argument $a(z)$:

```
> rminus, dr, rplus, P := SolverExactStableMPF(a):
```

The procedure SolverExactStableMPF returns the factors rminus, dr, rplus for the right factorization and permutation matrix P used for normalization. The package ExactStableMPF performs the same input/output tests as ExactMPF.

Let us illustrate the package by several examples. We will not provide a worksheet but will only present the result of the work of the package.

Example. Since the matrix polynomial $b(z)$ has stable right partial indices, we may apply the package ExactStableMPF

```
> rminus,dr,rplus,P:=SolverExactStableMPF(b):
```

"Version from 08.04.2025"

. . .

"The stable factorization of II-type..." . . .

"Normalization of factorization ..."

. . .

"The stable factorization is successfully constructed"

"The total computation time is 8.797 seconds"

```
> dr,P;
```

$$\begin{bmatrix} z^3 & 0 & 0 & 0 & 0 \\ 0 & z^3 & 0 & 0 & 0 \\ 0 & 0 & z^4 & 0 & 0 \\ 0 & 0 & 0 & z^4 & 0 \\ 0 & 0 & 0 & 0 & z^4 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Compared with ExactMPF, the execution time has been reduced by almost 10 times.

In the following example, we present the full output of the package.

Example. Let

$$c(z) = \begin{pmatrix} 37z^2+14z-14 & 71z^2+37z-28 & -z^2+3z \\ -z^2-13z-15 & z^2+13z+15 & z^2+13z+15 \\ z & z^2+2z & z^2 \end{pmatrix}.$$

After executing the package SolverExactStableMPF, we obtain the P-normalized stable factorization of II-type

$$c(z) = \begin{pmatrix} \frac{2527}{37782z} & 1-\frac{13349}{37782z} & 0 \\ 1+\frac{62381}{75564z} & \frac{109829}{75564z} & 0 \\ -\frac{361}{75564} & \frac{1907}{75564} & 1 \end{pmatrix} \cdot \begin{pmatrix} z & 0 & 0 \\ 0 & z & 0 \\ 0 & 0 & z^2 \end{pmatrix} \cdot \begin{pmatrix} -z-\frac{138434}{2099} & z-\frac{191053}{2099} & z+\frac{28605}{2099} \\ 37z+\frac{56966}{2099} & 71z+\frac{130177}{2099} & -z+\frac{5415}{2099} \\ -\frac{1970}{2099} & -\frac{1652}{2099} & \frac{2162}{2099} \end{pmatrix}.$$

and the permutation matrix $P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, that determines the type of normalization. The execution time was 0.110 seconds.

Let us verify that rminus has the form (3.9):

$$P^{-1} \cdot rminus = \left(\begin{array}{cc|c} 1+\frac{62381}{75564z} & \frac{109829}{75564z} & 0 \\ \frac{2527}{37782z} & 1-\frac{13349}{37782z} & 0 \\ \hline -\frac{361}{75564} & \frac{1907}{75564} & 1 \end{array}\right)$$

Finally we construct the Birkhoff factorization of $c(z)$ by its P-normalized Wiener–Hopf factorization. By Theorem 3.3 we have:

$$D_b = P \cdot d_r \cdot P^{-1}; \quad B_- = P \cdot d_r^{-1} \cdot P^{-1} \cdot rminus \cdot d_r; \quad B_+ = rplus.$$

Thus,

$$D_b = d_r; \; B_-(z) = \begin{pmatrix} \frac{2527}{37782z} & 1-\frac{13349}{37782z} & 0 \\ 1+\frac{62381}{75564z} & \frac{109829}{75564z} & 0 \\ -\frac{361}{75564z} & \frac{1907}{75564z} & 1 \end{pmatrix}$$

We can verify that $c(z) = D_b(z)B_-(z)B_+(z)$.

6. Package CanonicalSNSF

In the package CanonicalSNSF we consider only the right canonical factorizations such that $r_-(\infty) = I_p$.

The package CanonicalSNSF is designed to construct a canonical factorization with guaranteed accuracy for strictly non-singular matrix functions of the form

$$A(t) = \begin{pmatrix} 1 & \beta(t) \\ \alpha(t) & 1+\alpha(t)\beta(t) \end{pmatrix}, \tag{6.1}$$

where $\alpha(t) = \sum_{k=-\infty}^{\infty} \alpha_k t^k \in W$, $\beta(t) = \sum_{k=-\infty}^{\infty} \beta_k t^k \in W$. Additionally, we can assume that $\alpha_k, \beta_k \in \mathbb{Q}(i)$. We will use the new approach to the construction of a canonical factorization described in Sec. 2, Obstacle IV. Theoretical background for the package can be found in [7] and [17].

First, we will outline the relevant theory. We will use the notation $\alpha_M^N(z) = \sum_{k=M}^{N} \alpha_k z^k$, $-\infty \le M < N \le \infty$.

Let us represent $\alpha(t) = \alpha_{-\infty}^{-1}(t) + \alpha_0^{\infty}(t)$, $\alpha_{-\infty}^{-1}(t) \in W_-^0$, $\alpha_0^{\infty}(t) \in W_+$, and $\beta(t) = \beta_{-\infty}^0(t) + \beta_1^{\infty}(t)$, $\beta_{-\infty}^0(t) \in W_-$, $\beta_1^{\infty}(t) \in W_+^0$. Then

$$A(t) = \begin{pmatrix} 1 & 0 \\ \alpha_{-\infty}^{-1}(t) & 1 \end{pmatrix} a(t) \begin{pmatrix} 1 & \beta_1^{\infty}(t) \\ 0 & 1 \end{pmatrix},$$

where $a(t) = \begin{pmatrix} 1 & \beta_{-\infty}^0 \\ \alpha_0^{\infty} & 1 + \alpha_0^{\infty}\beta_{-\infty}^0 \end{pmatrix}$. As the class $\mathscr{E}F^{2\times 2} \subset GW^{2\times 2}$ (see in Obstacle IV for the concepts of this class) we take the set of the matrices of the form

$$A_N(t) = \begin{pmatrix} 1 & 0 \\ \alpha_{-\infty}^{-1}(t) & 1 \end{pmatrix} a_N(t) \begin{pmatrix} 1 & \beta_1^{\infty}(t) \\ 0 & 1 \end{pmatrix}, \tag{6.2}$$

where $a_N(t) = \begin{pmatrix} 1 & \beta_{-N}^0 \\ \alpha_0^N & 1 + \alpha_0^N\beta_{-N}^0 \end{pmatrix}$.

The Laurent matrix polynomial $a_N(t)$ admits an exact Wiener–Hopf factorization since $\det a_N(t) = 1$. Hence the class $\mathscr{E}F^{2\times 2}$ consists of matrix function admitting the exact factorization. Note that in this class the stable factorization coincides with the canonical one.

The exact factorization of $a_N(t)$ can be constructed with the help of the package ExactMPF or ExactStableMPF. For large N, the package ExactStableMPF allows one to check whether $a_N(t)$ admits an exact stable factorization and to build it much faster than ExactMPF. However, a canonical factorization of $a_N(t)$ can be constructed more simply without using the notions of indices and essential polynomials. This was done in the work [17]. Below we present these results.

6.1. An explicit canonical factorization of $a_N(t)$.

Let us introduce the matrix

$$D_N = I_N + \begin{pmatrix} \alpha_N & \dots & \alpha_1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \alpha_N \end{pmatrix} \begin{pmatrix} \beta_{-N} & \dots & 0 \\ \vdots & \ddots & \vdots \\ \beta_{-1} & \dots & \beta_{-N} \end{pmatrix}.$$

The matrix polynomial $a_N(t)$ admits the right canonical factorization if and only if the matrix D_N is invertible. Let D_N be an invertible matrix and $H_N = \big(h_{i,j}^{(N)}\big)_{i,j=0}^{N-1}$ is its inverse. Denote $H_k^{(N)}$, $0 \le k \le N-1$, the k-th column of H_N and let $H_k^{(N)}(t) = \sum_{j=0}^{N-1} h_{jk}^{(N)} t^j$ be the generating polynomial of this column. Denote $H_N^{(N)}(t) := t^N$ and $H_\alpha^{(N)}(t) := \sum_{k=0}^{N} \alpha_k H_k^{(N)}(t)$.

The right canonical factorization of $a_N(t)$, $a_N(t) = a_N^-(t)a_N^+(t)$, normalized by the condition $a_N^-(\infty) = I_2$, is constructed by the formula ([17], Theorem 7.1):

$$a_N^-(t) = \begin{pmatrix} 1 - \mathscr{P}_-\big(\beta_{-N}^0(t)H_\alpha^{(N)}(t)\big) & \mathscr{P}_-\big(\beta_{-N}^0(t)H_0^{(N)}(t)\big) \\ -\mathscr{P}_-\big(\beta_{-N}^0(t)\alpha_0^N(t)H_\alpha^{(N)}(t)\big) & 1 + \mathscr{P}_-\big(\beta_{-N}^0(t)\alpha_0^N(t)H_0^{(N)}(t)\big) \end{pmatrix},$$

$$a_N^+(t) = \begin{pmatrix} H_0^{(N)}(t) & \mathscr{P}_+\big(\beta_{-N}^0(t)H_0^{(N)}(t)\big) \\ H_\alpha^{(N)}(t) & 1 + \mathscr{P}_+\big(\beta_{-N}^0(t)H_\alpha^{(N)}(t)\big) \end{pmatrix}. \tag{6.3}$$

Remark 1. Let $\rho := \dim\ker D_N$. In the work [12, Corollary 1], it is proved that $(-\rho, \rho)$ are the right partial indices of $a_N(t)$. However, in this work formulas for the factors of $a_N(t)$ were not obtained.

Now we apply the approach of Sec.3 to the construction of an approximate canonical factorization of $A(t)$.

6.2. An estimate of $\|A - A_N\|_W$.

We will use the norms $\|\cdot\|_{W,1}$ and $\|\cdot\|_{W,\infty}$. If an estimate is valid for both $\|\cdot\|_{W,1}$ and $\|\cdot\|_{W,\infty}$, we will use the notation $\|\cdot\|_W$. Since

$$A(t) - A_N(t) = \begin{pmatrix} 1 & 0 \\ \alpha_{-\infty}^{-1} & 1 \end{pmatrix} (a(t) - a_N(t)) \begin{pmatrix} 1 & \beta_1^{\infty} \\ 0 & 1 \end{pmatrix},$$

then $\|A - A_N\|_W \le (1 + \|\alpha_{-\infty}^{-1}\|_W)(1 + \|\beta_1^\infty\|_W)\|a - a_N\|_W$. For $\|a - a_N\|_W$ we have:

$$\|a - a_N\|_W = \left\| \begin{pmatrix} 0 & \beta_{-\infty}^0 \\ \alpha_{N+1}^\infty & \alpha_0^\infty \beta_{-\infty}^0 - \alpha_0^N \beta_{-N}^0 \end{pmatrix} \right\|_W \le \|\beta_{-\infty}^0\|_W + \|\alpha_{N+1}^\infty\|_W +$$
$$\|\alpha_0^\infty \beta_{-\infty}^0 - \alpha_0^N \beta_{-N}^0\|_W \le \big(1 + \|\alpha_0^\infty\|_W\big)\|\beta_{-\infty}^{-N-1}\|_W + \big(1 + \|\beta_{-N}^0\|_W\big)\|\alpha_{N+1}^\infty\|_W \le$$
$$\big(1 + \|\alpha_0^\infty\|_W\big)\|\beta_{-\infty}^{-N-1}\|_W + \big(1 + \|\beta_{-\infty}^0\|_W\big)\|\alpha_{N+1}^\infty\|_W.$$

The final estimate is

$$\|A - A_N\|_W \le \delta_N := \big(1 + \|\alpha_{-\infty}^{-1}\|_W\big)\big(1 + \|\beta_1^\infty\|_W\big)\cdot$$
$$\Big[\big(1 + \|\alpha_0^\infty\|_W\big)\|\beta_{-\infty}^{-N-1}\|_W + \big(1 + \|\beta_{-\infty}^0\|_W\big)\|\alpha_{N+1}^\infty\|_W\Big]. \tag{6.4}$$

6.3. **An estimate of $\|(A_N^-)^{-1}\|_W$, $\|(A_N^+)^{-1}\|_W$.** For a 2×2 matrix function $A(t)$ it is valid $\|A^\#\|_{W,1} = \|A\|_{W,\infty}$ and vice versa. Here $A^\#$ is the adjoint matrix. Since $\det A_N^\pm(t) = 1$, we have $\|(A_N^\pm)^{-1}\|_{W,1} = \|A_N^\pm\|_{W,\infty}$ and $\|(A_N^\pm)^{-1}\|_{W,\infty} = \|A_N^\pm\|_{W,1}$.

Let $\det D_N \ne 0$. Using (6.2), we get

$$\|(A_N^-)^{-1}\|_{W,1} \le \big(1 + \|\alpha_{-\infty}^{-1}\|_W\big)\|a_N^-\|_{W,\infty}, \quad \|(A_N^+)^{-1}\|_{W,1} \le \big(1 + \|\beta_1^\infty\|_W\big)\|a_N^+\|_{W,\infty}.$$
$$\|(A_N^-)^{-1}\|_{W,\infty} \le \big(1 + \|\alpha_{-\infty}^{-1}\|_W\big)\|a_N^-\|_{W,1}, \quad \|(A_N^+)^{-1}\|_{W,\infty} \le \big(1 + \|\beta_1^\infty\|_W\big)\|a_N^+\|_{W,1}.$$

Here $a_N^\pm(t)$ are found by (6.3).

The following estimates follow from the work [17, Sec.7]:

$$\|a_N^-\|_W \le 1 + \|\beta_{-N}^0\|_W\big(1 + \|\alpha_0^N\|_W\big)^2 \max\{\|H_N\|_1, 1\}.$$
$$\|a_N^+\|_W \le 1 + \big(1 + \|\beta_{-N}^0\|_W\big)\big(1 + \|\alpha_0^N\|_W\big) \max\{\|H_N\|_1, 1\}.$$

Now we have

$$\|(A_N^-)^{-1}\|_W \le i(A_N^-) := \big(1 + \|\alpha_{-\infty}^{-1}\|_W\big)\Big[1 + \|\beta_{-N}^0\|_W\big(1 + \|\alpha_0^N\|_W\big)^2 \max\{\|H_N\|_1, 1\}\Big],$$
$$\|(A_N^+)^{-1}\|_W \le i(A_N^+) := \big(1 + \|\beta_1^\infty\|_W\big)\Big[1 + \big(1 + \|\beta_{-N}^0\|_W\big)\big(1 + \|\alpha_0^N\|_W\big) \max\{\|H_N\|_1, 1\}\Big].$$

6.4. **A criterion of the canonical factorization of $A(t)$.** Let us denote $i(A_N) := i(A_N^-)i(A_N^+)$. The following theorem is a consequence of Theorem 3.5.

Theorem 6.1. *Let there exist N such that the following conditions are fulfilled*

1. $\det D_N \ne 0$,
2. $q_N := \delta_N \cdot i(A_N) < 1$.

Then $A(t)$ admits the canonical factorization $A(t) = A_-(t)A_+(t)$.

We assume that this canonical factorization is normalized by the condition $A_-(\infty) = I_2$.

6.5. **An estimate of $\|A_\pm - A_N^\pm\|_W$.** We will need the following obvious estimate

$$\|A_N\|_W \le N(A_N) := (1 + \|\alpha_{-\infty}^{-1}\|_W)(1 + \|\beta_1^\infty\|_W)\big[1 + (1 + \|\alpha_0^N\|_W)(1 + \|\beta_{-N}^0\|_W)\big].$$

The following theorem follows from the results of the work [3].

Theorem 6.2. *Suppose that the conditions of Theorem 6.1 are fulfilled and $A_N(t) = A_N^-(t)A_N^+(t)$ is the canonical factorization of $A_N(t)$ constructed by the formulas (6.2), (6.3). Assume that we are looking for the canonical factorization $A(t) = A_-(t)A_+(t)$ of $A(t)$ normalized by the condition $A_-(\infty) = I_2$.*

Then

$$\|A_+ - A_N^+\|_W \le \frac{i^2(A_N)}{(1 - q_N)}\delta_N; \tag{6.5}$$

$$\|A_- - A_N^-\|_W \le \left[i(A_N^+) + \frac{i^2(A_N)}{(1 - q_N)}\Big(N(A_N) + \delta_N\Big)\right]\delta_N. \tag{6.6}$$

If, for some N, the conditions of Theorem 6.1 are fulfilled then there exists a number N_0 such that all matrix functions $a_N(z)$ and $A_N(z)$ admit the canonical factorization for all $N > N_0$. Then $\|A_\pm - A_\pm^{(N)}\|_W \to 0$ as $N \to \infty$. Hence for large enough N the approximate equality $A(t) \approx A_-^{(N)}(t)A_+^{(N)}(t)$ provides the approximate canonical factorization of $A(t)$ with the guaranteed accuracy.

6.6. **A description of the package CanonicalSNSF..** The input data for the package are the functions $\alpha(z)$, $\beta(z)$, for which the Fourier series expansions are known

$$\alpha(z) = \sum_{n=-\infty}^{\infty} \alpha_n z^n, \quad \beta(z) = \sum_{n=-\infty}^{\infty} \beta_n z^n, \quad |z| = 1.$$

We will represent $\alpha(z)$ and $\beta(z)$ as $\alpha(z) = \alpha_{-\infty}^{-1}(z) + \alpha_0^\infty(z)$ and $\beta(z) = \beta_{-\infty}^{-1}(z) + \beta_0^\infty(z)$. Let $\alpha_{-\infty}^{-1}(z) = \alpha_{-1}z^{-1} + \cdots + \alpha_{-N_1+1}z^{-N_1+1} + \sum_{n=-\infty}^{-N1} f_1(n)z^n$. We will pass information about this series to the package as the list

$$l1 := \left[\alpha_{-1}z^{-1} + \cdots + \alpha_{-N_1+1}z^{-N_1+1}, -N_1, f_1(n)z^n\right].$$

Thus, we can recover $\alpha_{-\infty}^{-1}(z)$ by the list $l1$ as follows: $\alpha_{-\infty}^{-1}(z) = l1[1] + \sum_{n=-\infty}^{l1[2]} l1[3]$. Analogously, the list $l2 := \left[\alpha_0 + \cdots + \alpha_{N_2-1}z^{N_2-1}, N_2, f_2(n)z^n\right]$ will correspond to the series $\alpha_0^\infty(z) = l2[1] + \sum_{n=l2[2]}^{\infty} l2[3]$. The lists $l3$ and $l4$ for the function $\beta(z)$ are defined similarly.

Let us also define the parameters epsilon $= 10^{-\delta}$ and $Nmax > Nmin := \max\{|l1[2]|, l2[2], |l3[2]|, l4[2]\} + 1$. The parameter epsilon is the accuracy with which the factors $A_\pm(z)$ are to be found. The package CanonicalSNSF checks whether there exists a number $Ncrit \in [Nmin, Nmax]$ for which the conditions of the canonical factorization of $A(t)$ are fulfilled and the number $Napprox \in [Nmin, Nmax]$ that guarantees the required accuracy epsilon. If these numbers do not exist, the package suggests to increase $Nmax$ and then performs the calculations again. If this does not help, then it is highly likely that $A(t)$ does not admit the canonical factorization.

To access CanonicalSNSF use the commands

```
> read("CanonicalSNSF.txt");
> with(CanonicalSNSF);
> with(LinearAlgebra);
```

To obtain the factorizations of $A(t)$ we run SolverCanonicalSNSF with the arguments l1,l2,l3,l4,epsilon:

```
>N,rminus,rplus:= SolverCanonicalSNSF(l1,l2,l3,l4,epsilon):
```

The procedure SolverCanonicalSNSF returns the number N=Napprox and the factors rminus, rplus for the right approximate canonical factorization of $A(t)$ with guaranteed accuracy epsilon.

Example. Check that the matrix function

$$A(t) = \begin{pmatrix} 1 & e^{\frac{2}{z}} \\ e^{2z} & 1 + e^{(\frac{2}{z}+2z)} \end{pmatrix}$$

admits the canonical factorization and construct it with the accuracy $\varepsilon = 10^{-3}$.

Solution. Here $\alpha(z) = e^{2z} = 0 + \sum_{n=0}^{\infty} \frac{2^n z^n}{n!}$, and $\beta(z) = e^{\frac{2}{z}} = \sum_{n=-\infty}^{-1} \frac{2^{-n}z^n}{(-n)!} + 1$. Let us form the input data for package CanonicalSNSF. The series $\alpha(z)$, $\beta(z)$ are are represented by the lists:

$$l1 := [0, -1, 0], \; l2 := [0, 0, \frac{2^n z^n}{n!}], \quad l3 := [0, -1, \frac{2^{-n}z^n}{(-n)!}], \; l4 := [1, 1, 0].$$

Here Nmin $:= \max(\text{abs}(l1[2]), l2[2], \text{abs}(l3[2]), l4[2]) + 1 = 2$. Put Nmax $:= 10$ and apply SolverCanonicalSNSF.

```
> N,Am,Ap:=SolverCanonicalSNSF(l1,l2,l3,l4,epsilon,Nmax):
```

"Version from 20.12.2025"
"The criterion of the canonical factorization is not fulfilled. Increase Nmax"
"If this trick does not help, then it is highly likely that the canonical factorization of A(t) does not exist"
"The total computation time is .750 seconds"

Thus, for $N \in [2, 10]$ the conditions of Theorem 6.1 are not fulfilled. Let us increase Nmax: Nmax := 20. Now we have

```
> N,Am,Ap:=SolverCanonicalSNSF(l1,l2,l3,l4,epsilon,Nmax):
```

"Version from 20.12.2025"
"The required accuracy has not been achieved. Nmax must be increased"
"The total computation time is .844 seconds"

Put Nmax := 30.

```
> N,Am,Ap:=SolverCanonicalSNSF(l1,l2,l3,l4,epsilon,Nmax):
```

"Version from 20.12.2025"
"SolverExactSNSF. Version from 13.06.2025"
"The canonical factorization of a_N was completed successfully"
"The total computation time is 7.031 seconds"

```
>N, Determinant(Am), Determinant(Ap);
```

23, 1, 1

Conclusion

In this work, we have taken the first fundamental step towards constructing an Automated Research System for the Wiener–Hopf factorization problem. We have created three packages: ExactMPF, ExactStableMPF, and CanonicalSNSF.

Future development of the ARS will be based on the systematic application of the new approach to constructing the stable factorization for various classes of matrix functions. The class of Laurent matrix polynomials will be chosen as $\mathscr{E}F^{p\times p}$ and the package ExactMPF or ExactStableMPF will be used for factorization within this class.

First of all, this approach is supposed to be applied to the class of matrix polynomials, without the strict requirement of their exact factorizability. Subsequently, the approach will be extended to the class of rational matrix functions and the Khrapkov class. In addition, further development of the package CanonicalSNSF is required to make the specification of input data more user-friendly.

Author contributions

All authors contributed equally to the development of the theory, the design of the packages and the preparation of the manuscript.

Acknowledgement

Victor Adukov gratefully acknowledges the Russian Science Foundation, grant No 25-21-20134.

The authors are grateful to the reviewers for valuable comments and constructive suggestions that significantly improved the exposition and presentation of the paper.

References

1. I. D. Abrahams: *On the application of the Wiener–Hopf technique to problems in dynamic elasticity.* Wave Motion **36** (2002), 311–333. https://doi.org/10.1016/S0165-2125(02)00027-6.
2. I. D. Abrahams, X. Huang, A. V. Kisil, G. Mishuris, M. Nieves, S.V. Rogosin, I. Spitkovsky: *Reinvigorating the Wiener–Hopf technique in the pursuit of understanding processes and materials.* National Science Review **13** issue 2 (2021), ID nwaa225. https://doi.org/10.1093/nsr/nwaa225
3. N. V. Adukova, V. L. Dilman: *Stability of factorization factors of the canonical Wiener-Hopf factorization of matrix functions.* Bull. South Ural Univ. Series Math., Mech., Physics. 2021, **13** no. 1 (2021), 5–13. https://doi.org/10.14529/mmph210101
4. N. V. Adukova: *Stability of the factors in the Wiener–Hopf factorization of matrix functions.* Bull. South Ural State Univ. Series Math., Mech., Physics. **14** no. 1 (2022), 5–12. https://doi.org/10.14529/mmph220101
5. N. V. Adukova, V. M. Adukov: *On effective criterion of stability of partial indices for matrix polynomials.* Proc. R. Soc. A **476** issue 2238 (2020), ID 20200012. http://dx.doi.org/10.1098/rspa.2020.0012.
6. N. V. Adukova, V. M. Adukov: *On a normalization of the Wiener–Hopf factorization for matrix functions.* Book of abstract of the 13th ISAAC Congress. Ghent, Belgium. 45,(2021).
7. N. V. Adukova, V. M. Adukov, G. Mishuris: *An effective criterion for a stable factorization of strictly nonsingular* 2×2 *matrix functions: use of the ExactMPF package.* Proc. R. Soc. A **480** issue 2299 (2024), ID 20240116. https://doi.org/10.1098/rspa.2024.0116.

8. V. M. Adukov, N. V. Adukova, G. Mishuris: *An explicit Wiener–Hopf factorization algorithm for matrix polynomials and its exact realizations within ExactMPF package.* Proc. R. Soc. A **478** issue 2263 (2022), ID 20210941. https://doi.org/10.1098/rspa.2021.0941
9. V. M. Adukov, N.V. Adukova, G. Mishuris: *Note on exact factorization algorithm for matrix polynomials.* Bull. South Ural State Univ. Series Math. modeling, Programming and Computer Software. **16** no. 1 (2023), 47–58. https://doi.org/10.14529/mmp230104
10. V. M. Adukov, G. Mishuris: *Utilization of the ExactMPF package for solving a discrete analogue of the nonlinear Schrödinger equation by inverse scattering transform method.* Proc. R. Soc. A **479** issue 2269 (2023), ID 20220144. https://doi.org/10.1098/rspa.2022.0144.
11. V. M. Adukov: *Wiener–Hopf factorization of meromorphic matrix-valued functions.* St. Petersburg Math. J. **4** no. 1 (1993), 51–69.
12. V. M. Adukov: *On factorization indices of strictly nonsingular* 2×2 *matrix functions.* Integr. Equat. Oper. Th. **21** no. 1 (1995), 1–11. https://doi.org/10.1007/BF01262988
13. V. M. Adukov: *Generalized inversion of block Toeplitz matrices.* Linear Algebra Appl. **274**(1–3) (1998), 85–124. https://doi.org/10.1016/S0024-3795(97)00304-2.
14. V. M. Adukov: *Factorization of analytic matrix-valued functions.* Theor. Math. Physics. bf 118 no.3 (1999), 255–263. https://doi.org/10.4213/tmf704.
15. V. M. Adukov: *Normalization of Wiener–Hopf factorization for* 2×2 *matrix functions and its application.* Ufa Math. J. **14** no. 4 (2022), 1–13. https://doi.org/10.13108/2022-14-4-1
16. V. M. Adukov: *Normalization of Wiener–Hopf factorization for matrix functions with distinct partial indices.* Trans. Razmadzen Math. Inst. **176** issue 3 (2022), 313–321.
17. V. M. Adukov: *A Method of Essential Functions and Its Application in the Theory of Wiener–Hopf Factorization.* Ufa Math. J., accepted for publication.
18. G. D. Birkhoff: *A theorem on matrices of analytic functions.* Math. Ann. **74** (1913), 122–133.
19. B. V. Bojarski: *On stability of Hilbert problem for holomorphic vector.* Soobsch. AN Gruz. SSR **21** no. 4 (1958), 391–398.
20. K. F. Clancey, I. C. Gohberg: *Factorization of matrix functions and singular integral operators (Operator Theory, Advances and Applications).* **3** Springer, Basel (1981). https://doi.org/10.1007/978-3-0348-5492-4.
21. V. G. Daniele, R. C. Zich: *The Wiener–Hopf method in electromagnetics.* ISMB Series. SciTech Publishing, Edison, New York (2014). https://doi.org/10.1049/SBEW503E.
22. L. Ephremidze, F. Saied, I. M. Spitkovsky: *On the Algorithmization of Janashia–Lagvilava Matrix Spectral Factorization Method.* IEEE Transactions on Information Theory **64** no. 2 (2018), 728–737. https://doi.org/10.1109/TIT.2017.2772877
23. L. Ephremidze, I. M. Spitkovsky: *On Explicit Wiener–Hopf Factorization of* 2×2 *Matrices in a Vicinity of a Given Matrix.* Proceedings of the Royal Society A **476** (2020), Article 20200027. https://doi.org/10.1098/rspa.2020.0027
24. L. Ephremidze: *Algorithm for Wiener–Hopf Factorization of Polynomial Matrices with Monomial Determinant.* Journal of Mathematical Sciences **290** (2025), 693–701. https://doi.org/10.1007/s10958-025-07685-5
25. I. C. Gohberg, I. A. Feldman: *Convolution equations and projection method for their solution.* Amer. Math. Soc. Transl. Math. Monographs **41** Providence, RI (1974).
26. I. C. Gohberg, M. G. Krein M.G: *Systems of integral equations on a half-line with kernels depending on the difference of arguments.* Uspekhi Mat. Nauk **13** (1958), 3–72.
27. I. C. Gohberg, M. G. Krein: *On the stable system of partial indices of the Hilbert problem for many unknown functions.* Dokl. Akad. Nauk SSSR **119** no. 5 (1958), 854–857.
28. I. C. Gohberg, L. Lerer, L. Rodman: *Factorization indices for matrix polynomials.* Bull. Amer. Math. Soc. **84** no. 2 (1978), 275–277. https://doi.org/0.1090/S0002-9904-1978-14473-5
29. G. Giorgadze, N. Manjavidze: *On some constructive methods for the matrix Riemann–Hilbert boundary value problem.* J. Math. Sci. **195** no. 2 (2013), 146–174. https://doi.org/10.1007/s10958-013-1571-7.
30. R. A. Horn, C. R. Johnson: *Matrix Analysis.* Cambridge University Press, Cambridge-London-NY. (1986).
31. G. Janashia, E. Lagvilava, L. Ephremidze: *A New Method of Matrix Spectral Factorization.* IEEE Transactions on Information Theory **57** no. 4 (2011), 2318–2326. https://doi.org/10.1109/TIT.2011.2112233
32. A. V. Kisil, I. D. Abrahams, G. Mishuris, S.V. Rogosin: *The Wiener–Hopf technique, its generalizations and applications: constructive and approximate methods.* Proc. R. Soc. A **477** issue 2254 (2021), ID 20210533. https://doi.org/10.1098/rspa.2021.0533
33. J. B. Lawrie, I. D. Abrahams: *A brief historical perspective of the Wiener–Hopf technique.* J. Eng. Math. **59** (2007), 351–358. https://doi.org/10.1007/s10665-007-9195-x.
34. G. S. Litvinchuk, I. M. Spitkovsky: *Factorization of measurable matrix functions.* Birkhäuser, Basel-Boston (1987). https://doi.org/10.1515/9783112471722
35. N. I. Muskhelishvili: *Singular integral equations.* Nauka, Moscow (1968).
36. S. Prössdorf: *Some classes of singular equations.* Series North-Holland mathematical library, **17** North-Holland Pub. Co, Amsterdam, New York (1978).
37. S. V. Rogosin, G. Mishuris: *Constructive methods for factorization of matrix-functions.* IMA J. Appl. Math. **81** no. 2 (2016), 365–391. https://doi.org/10.1093/imamat/hxv038.

38. M. A. Shubin: *Factorization of parameter-dependent matrix-functions in normed rings and certain related questions in the theory of Noether operators.* Math. USSR-Sb. **2** no. 4 (1967), 543–560.
39. F-O Speck: *General Wiener-Hopf Factorization Methods.* Research Notes in Mathematics **119** AMS (1985).
40. N. P. Vekua: *Systems of singular integral equation and some boundary value problems.* Nauka, Moscow (1970).

(Received ??.??.20??)

[1]Department of Mathematics, Aberystwyth University, Penglais, Aberystwyth, Ceredigion, UK SY23 3FL

[2]Institute of Natural Sciences and Mathematics, South Ural State University, 76, Lenin prospekt, Chelyabinsk, Russia, 454080

[3]Department of Mathematics, Aberystwyth University, Penglais, Aberystwyth, Ceredigion, UK SY23 3FL

Email address: naa53@aber.ac.uk
Email address: adukovvm@susu.ru
Email address: ggm@aber.ac.uk