On explicit and exact solutions of the Wiener-Hopf factorization problem for some matrix functions

Victor Adukov

South Ural State University, Russia

WHT2019, 12.08–16.08

To present some theoretical considerations and algorithms necessary to design solvers for explicit solving the Wiener–Hopf factorization problem.

Requirements for algorithms. To implement the factorization algorithms, the following requirements must be fulfilled:

R1. a solution of the problem can be defined by a finite numbers of numerical parameters,

R2. an initial data is a finite numerical set,

R3. a finite number of machine operations are required to find the solution.

We want to present classes of matrix function and algorithms for which these requirements are realized.

Exact algorithm. Since the problem is unstable, it is highly desirable that the factorization algorithm would give an exact result if the initial data is exact. We say that such algorithn is exact.

- It was demonstrated that the problem can be explicitly solved for matrix Laurent polynomials, rational matrix functions, analytic matrix functions, and meromorphic matrix functions.
- It was shown that for triangular 2×2 matrix functions the partial indices can be explicitly found.
  The factorization of triangular matrix functions of arbitrary order was reduced to analytic case with the help of some recurrent procedure.
- It was developed exact factorization algorithms for these classes of matrix functions.
- It was designed a solver for exact factorization of matrix polynomials.

- Γ is a simple smooth closed contour in $\mathbb{C}$.
- D_+ is the interior domain, $0 \in D_+$. $D_- = \overline{\mathbb{C}} \setminus (D_+ \cup \Gamma)$.
- $a(t)$ is a continuous and invertible $p \times p$ matrix function on Γ.
- A right Wiener–Hopf factorization of $a(t)$:

$$a(t) = r_-(t) d_r(t) r_+(t), \ t \in \Gamma. \qquad (1)$$

$r_\pm(t)$ are continuous and invertible on $\overline{D}_\pm$, analytic into $D_\pm$;
$d_r(t) = \mathrm{diag}\,[t^{\rho_1}, \dots, t^{\rho_n}]$, $\rho_1 \leq \dots \leq \rho_p$, $\sum \rho_j = \varkappa := \mathrm{ind}\,\det a(t)$.

- A left Wiener-Hopf factorization of $a(t)$:

$$a(t) = l_+(t) d_l(t) l_-(t), \ t \in \Gamma, \qquad (2)$$

$d_l(t) = \mathrm{diag}[t^{\lambda_1}, \dots, t^{\lambda_p}]$, $\lambda_1 \geq \dots \geq \lambda_p$.

Remark

An important feature: we will simultaneously use both factorizations.

Assumptions to simplify the design of the solvers

A1. Γ is the unit circle $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$.

A2. All functions $f(z)$ in the problem must be analytic in some annular domain $r \leq |z| \leq R$, $0 \leq r < 1 < R \leq \infty$. Hence $f(z) = \sum_{j=-\infty}^{\infty} f_j z^j$.

Remarks

- *The factorization problem on the contour Γ can be transformed to the problem on $\mathbb{T}$ by a suitable conformal mapping.*
- *As a rule, functions appearing in applications are analytic into some neighborhood of the contour $\mathbb{T}$. To define them we can*
  (i) *set a library of analytic functions,*
  (ii) *use expansions in the Laurent series.*

What is an explicit solution in the literature?

A solution of the factorization problem is **explicit** if

- there exist analytic formulas for a solution of the problem (scalar functions, functionally commutative matrix functions, matrix functions are close to the identety matrix),
- or the factorization is reduced to the scalar case, (2×2 triangular matrix functions and matrix functions that reduced to them),
- or the solution can be obtained by constructive (effective) methods.

Remark

The last means that the solution can be found by a finite number of some steps which we call "explicit". When we solve a specific factorization problem we must rigorously define these steps.

What are explicit solutions in this talk?
A solution of the factorization problem is **explicit** if

E1. the factors $l_+(z)$ and/or $l_-(z)$ ($r_+(z)$ and/or $r_-(z)$) are determined by a finite number of numerical parameters;

E2. it is defined by a finite number of numerical initial data;

E3. it can be constructed from the initial data by a finite number of basic matrix operations and solving systems of linear equations.

Examples

- *For an analytic scalar function a solution is explicit,*
- *For a general scalar function a solution is not explicit. since E1 does not fulfill.*

$l_+(z)$ and/or $l_-(z)$ ($r_+(z)$ and/or $r_-(z)$) are rational matrix functions $\Rightarrow E1$.

Special cases:

MP_m^n: $l_+(z)$ and $l_-(z)$ ($r_+(z)$ and $r_-(z)$) are matrix polynomials in z and in z^{-1} $\Leftrightarrow a(z) = \sum_{j=-m}^{n} a_j z^j$.

RMF: $l_+(z)$ and $l_-(z)$ ($r_+(z)$ and $r_-(z)$) are rational matrix functions $\Leftrightarrow a(z)$ is a rational matrix function.

$AMF_m^{\pm}$: $l_-(z)$ or $l_+(z)$ ($r_-(z)$ or $r_+(z)$) is a matrix polynomial in $z^{\mp 1} \Leftrightarrow a(z) = \sum_{j=-m}^{+\infty} a_j z^j$ or $a(z) = \sum_{j=-\infty}^{m} a_j z^j$.

$MMF^{\pm}$: $l_-(z)$ or $l_+(z)$ ($r_-(z)$ or $r_+(z)$) is rational matrix function $\Leftrightarrow a(z)$ is a meromorphic in D_+ or D_- matrix function.

Remark

The basic classes: MP_0^n, AMF_0^+.
The factorization of the rest can be reduced to basic classes.

E2. The initial data will be a finite sequence $c_M^N = \{c_M, c_{M+1}, \ldots, c_N\}$, $M < N$, of the Laurent coefficients for some $p \times p$ matrix functions.
E3. The factorizations will be constructed in terms of indices and essential polynomials of c_M^N in a finite number of steps.

Definition of indices of c_M^N.
1. Form the sequence T_k,

$$T_k = \begin{pmatrix} c_k & c_{k-1} & \cdots & c_M \\ c_{k+1} & c_k & \cdots & c_{M+1} \\ \vdots & \vdots & & \vdots \\ c_N & c_{N-1} & \cdots & c_{N+M-k} \end{pmatrix}, \quad M \leq k \leq N.$$

Definition

*c_M^N is **regular** $\overset{def}{\Longleftrightarrow}$ T_M, T_N have the full rank.*

2. Form the sequence $\Delta_k = d_k - d_{k-1}$, $d_k := \dim\ker T_k$, $d_{M-1} := 0$, $d_{N+1} := (N-M+2)p$, $M \le k \le N+1$.

Theorem

c_M^N is regular $\Rightarrow$
$\exists$ integers $\mu_1 \le \cdots \le \mu_{2p}$ such that

$$\begin{array}{ccccccc} \Delta_M & = & \cdots & = & \Delta_{\mu_1} & = & 0, \\ \dots & & & & & & \dots \\ \Delta_{\mu_i+1} & = & \cdots & = & \Delta_{\mu_{i+1}} & = & i, \\ \dots & & & & & & \dots \\ \Delta_{\mu_{2p}+1} & = & \cdots & = & \Delta_{N+1} & = & 2p. \end{array} \qquad (3)$$

The absence of the jth row here means that $\mu_j = \mu_{j+1}$.

Definition

$\mu_1, \dots, \mu_{2p}$ are indices of c_M^N.

An important feature: we will simultaneously use the right and left kernels of T_k.

The notation: $\mathcal{N}_k^R$ ($\mathcal{N}_k^L$) is the space of generating column (row) polynomials in z (z^{-1}) for vectors from $\ker_R T_k$ ($\ker_L T_k$).

Theorem (structure of $\mathcal{N}_k^{R,L}$)

c_M^N is regular, $\varkappa_j$ is the multiplicity of the index μ_j, $1 \leq j \leq 2p$, $\Rightarrow$

$$\mathcal{N}_k^R,\ z\mathcal{N}_k^R \subseteq \mathcal{N}_{k+1}^R, \quad \mathcal{N}_{k-1}^L \supseteq \mathcal{N}_k^L,\ z^{-1}\mathcal{N}_k^L$$

$$\mathcal{N}_{k+1}^R = \mathcal{N}_k^R + z\mathcal{N}_k^R,\ \ \mathcal{N}_{k-1}^L = \mathcal{N}_k^L + z^{-1}\mathcal{N}_k^L,\ k \neq \mu_j,$$

$$\mathcal{N}_{\mu_j+1}^R = \left(\mathcal{N}_{\mu_j}^R + z\mathcal{N}_{\mu_j}^R\right) \oplus \mathcal{H}_{\mu_j+1}^R,$$

$$\mathcal{N}_{\mu_j-1}^L = \left(\mathcal{N}_{\mu_j}^L + z^{-1}\mathcal{N}_{\mu_j}^L\right) \oplus \mathcal{H}_{\mu_j-1}^L,\ \dim \mathcal{H}_{\mu_j\pm1}^{R,L} = \varkappa_j.$$

Definition of the right and left essential polynomials of c_{-M}^N

Definition

$\{R_j(z), \dots, R_{j+\varkappa_j-1}(z)\}$ is a basis of $\mathcal{H}^R_{\mu_j+1} \overset{def}{\Longleftrightarrow} R_j(z), \dots, R_{j+\varkappa_j-1}(z)$ are the right essential polynomials of c_M^N corresponding to μ_j.
$\{L_j(z), \dots, L_{j+\varkappa_j-1}(z)\}$ is a basis of $\mathcal{H}^L_{\mu_j-1} \overset{def}{\Longleftrightarrow} L_j(z), \dots, L_{j+\varkappa_j-1}(z)$ are the left essential polynomials of c_M^N corresponding to μ_j.

Result. For any regular c_M^N there exist the indices $\mu_1, \dots, \mu_{2p}$, the right essential polynomials $R_1(z), \dots, R_{2p}(z)$, the left essential polynomials $L_1(z), \dots, L_{2p}(z)$.
Operations that are needed to determinate the indices and essential polynomials:

- finding $\operatorname{rank} T_k$;
- solving the homogeneous systems $T_k X = 0$ and $Y T_k = 0$, $M \le k \le N$.

$a(z) \in MP_0^n$, $n := \deg a(z)$.

Initial data. c^m_{-m}, $\forall m \geq n$, $c_j = \frac{1}{2\pi i} \int_{|t|=1} t^{-j-1} a^{-1}(t)\, dt$.

Definition

Essential polynomials $R_1(z), \dots, R_{2p}(z)$, $L_1(z), \dots, L_{2p}(z)$ of c^m_{-m} are called ***factorization essential polynomials (FEP)*** *if*

- *the leading terms of $R_1(z), \dots, R_p(z)$ are equal to zero and*
- *the constant terms of $L_{p+1}(z), \dots, L_{2p}(z)$ are equal to zero.*

Theorem

c^m_{-m} is regular and has FEP. If $\mu_1, \dots, \mu_{2p}$ are indices and $R_1(z) \dots, R_{2p}(z)$, $L_1(z) \dots, L_{2p}(z)$ are FEP of c^m_{-m}, then $\lambda_j = -\mu_j$, $\rho_j = \mu_{p+j}$, $j = 1, \dots, p$,

$$l_+(z) = (R_1(z), \dots, R_p(z)), \;\; r_+(z) = z^{m+1} d_r^{-1}(z) \begin{pmatrix} L_{p+1}(z) \\ \vdots \\ L_{2p}(z) \end{pmatrix}.$$

Stability criterion of Gohberg–Krein–Bojarsky and the above theorem $\Rightarrow$

Theorem

Let $a(z) \in MP_0^n$, $n := \deg a(z)$, $\varkappa := \operatorname{ind} \det a(z)$, $\varkappa = \nu p + r$, $\nu \geq 0$, $0 \leq r < p$.

$$\lambda_1, \ldots, \lambda_p \quad (\rho_1, \ldots, \rho_p) \text{ are stable} \Leftrightarrow$$

- $\operatorname{rank} T_{-\nu-1}(c_{-n}^n) = (n-\nu)p \quad (\operatorname{rank} T_{\nu+1}(c_{-n}^n) = (n-\nu)p)$,
- $\operatorname{rank} T_{-\nu}(c_{-n}^n) = (n+1)p - \varkappa \quad (\operatorname{rank} T_{\nu}(c_{-n}^n) = (n+1)p - \varkappa)$.

Special cases.
$\nu = 0$: Stability $\Leftrightarrow \operatorname{rank} T_{-1}(c_{-n}^n) = np \quad (\operatorname{rank} T_1(c_{-n}^n) = np)$,
$r = 0$: Stability $\Leftrightarrow \operatorname{rank} T_{-\nu}(c_{-n}^n) = (n-\nu+1)p$
$(\operatorname{rank} T_{\nu}(c_{-n}^n) = (n-\nu+1)p)$.

Problem.

To prove the criterion independently of the Gohberg–Krein–Bojarsky theorem.

$a(z) \in AMF_0^+$, $\varkappa := \mathrm{ind}\det a(z)$.

Initial data. $c_{-m-\varkappa}^{m-\varkappa}$, $\forall m \geq \varkappa$, $c_j = \frac{1}{2\pi i}\int_{|t|=1} t^{-j-1} a^{-1}(t)\,dt$.

Definition

Essential polynomials $R_1(z), \dots, R_{2p}(z)$, $L_1(z), \dots, L_{2p}(z)$ of $c_{-m-\varkappa}^{m-\varkappa}$ are called ***factorization essential polynomials (FEP)*** *if*

- *the constant terms of $R_1(z), \dots, R_p(z)$ are equal to zero and*
- *the leading terms of $L_{p+1}(z), \dots, L_{2p}(z)$ are equal to zero.*

Theorem

$c_{-m-\varkappa}^{m-\varkappa}$ is regular and has FEP. If $\mu_1, \dots, \mu_{2p}$ are indices and $R_1(z) \dots, R_{2p}(z)$, $L_1(z) \dots, L_{2p}(z)$ are FEP of $c_{-m-\varkappa}^{m-\varkappa}$, then $\rho_j = \mu_j + 2\varkappa$, $\lambda_j = -\mu_{p+j}$, $j = 1, \dots, p$,

$$r_-(z) = z^{-m+\varkappa-1}\left(R_1(z), \dots, R_p(z)\right) d_r^{-1}(z),\ \ l_-(z) = \begin{pmatrix} L_{p+1}(z) \\ \vdots \\ L_{2p}(z) \end{pmatrix}.$$

Reduction to the factorization in MP_0^n or AMF_0^+:

- $a(z) \in MP_m^n$ or $AMF_m^+ \Rightarrow z^m a(z) \in MP_0^n$ or AMF_0^+,
- $a(z) \in RMP$ or $MMF^+ \Rightarrow \exists$ a scalar polynomial $q(z)$ such that $q(z)a(z) \in MP_0^n$ or AMF_0^+,
- $a(z) \in A_m^-$ or $MMF^- \Rightarrow a(z^{-1}) \in A_m^+$ or MMF^+.

Remark

To construct the factorization of a matrix polynomial or an analytic in D_+ matrix function $a(z)$ we can also use the sequence c_{-m}^{-1} for any $m \geq 2\varkappa$ or the sequence d_{-m}^{m}, $\forall m \geq \varkappa$.
Here d_j is the Laurent coefficient of $\Delta_-^{-1}(z)a(z)$,
$\Delta(z) = \Delta_-(z)z^{\varkappa}\Delta_+(z)$ is the factorization of $\Delta(z) = \det a(z)$.

An explicit solution – the revised definition

A solution of the factorization problem is **explicit** if

RE1. it is defined by a finite number of numerical initial data;

RE2. the factorization of scalar functions or the known factorization of certain matrix functions can be used to compute initial data;

RE3. it can be constructed from the initial data by a finite number of basic matrix operations, solving systems of linear equations, and applying the projector P_+.

Here $P_+(\sum_{j=-\infty}^{\infty} f_j z^j) = \sum_{j=0}^{\infty} f_j z^j$.

Remark

To implement an algorithm of this explicit factorization we will have

- *to use an approximate scalar factorization and*
- *to approximate $P_+ f(z)$ by a finite sum.*

$$A(z) = \begin{pmatrix} a_{11}(z) & 0 \\ a_{21}(z) & a_{22}(z) \end{pmatrix}, \; a_{jj}(z) = a_{jj}^{-}(z) z^{\nu_j} a_{jj}^{+}(z).$$

Only the right factorization of $A(z)$ is considered.

Theorem

$\nu_2 \le \nu_1 + 1 \;\Rightarrow$ the factorization can be constructed by elementary methods; ν_1, ν_2 are the right partial indices of $A(z)$.

Initial data for $\nu_2 \ge \nu_1 + 2$: $b_{\nu_1+1}^{\nu_2-1}$, $b_j = \frac{1}{2\pi i} \int_{|t|=1} t^{-j-1} \frac{a_{21}(t)}{a_{22}^{-}(t) a_{11}^{+}(t)} \, dt$.

Theorem

$\nu_2 \ge \nu_1 + 2$; μ_1, μ_2 are the indices of $b_{\nu_1+1}^{\nu_2-1} \;\Rightarrow \mu_1$, μ_2 are the right partial indices of $A(z)$.

Remark

The explicit factorization of $A(z)$ can be constructed in terms of the essential polynomials of $b_{\nu_1+1}^{\nu_2-1}$ and the function $P_+\left(z^{-\nu_1} \frac{a_{21}(z)}{a_{22}^{-}(z) a_{11}^{+}(z)}\right)$.

Only the right factorization is considered.

Triangular matrix functions.
Using the factorization of the diagonal elements of $A(z)$ and applying the operation P_+ recurrently, the factorization of $A(z)$ is reduced to the factorization of some analytic matrix function.

Block triangular matrix functions.

$$A(z) = \begin{pmatrix} A_{11}(z) & 0 \\ A_{21}(z) & A_{22}(z) \end{pmatrix}, \; A_{jj}(z) = A_{jj}^{-}(z) d_j(z) A_{jj}^{+}(z) \quad \Rightarrow$$

using $A_{jj}^{\pm}(z)$ and the operation P_+, the factorization of $A(z)$ is reduced to the factorization of some analytic matrix function.

$d_{jj}(z) = z^{\nu_j} I_p \quad \Rightarrow$ the factorization can be constructed explicitly via the indices and essential polynomials of the matrix sequence $b_{\nu_1+1}^{\nu_2-1}$.

Definition

*A solution of the factorization problem is **exact** if it is explicit and it can be found by symbolic computation.*

New assumptions

A1. The contour Γ is the unit circle $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$.

A2. All matrix functions $f(z)$ in the problem has the form $f(z) = \sum_{j=-\infty}^{\infty} f_j z^j$,

new: A3. $f_j \in \mathbb{Q}(i)$.

Remark

$A3 \;\Rightarrow$ *all calculations are realized in the exact arithmetic.*

Why do we need exact solutions?
They will be useful in numerical experiments when developing stable methods for the Wiener–Hopf factorization.

Algorithm 1. The exact factorization of $p(z) \in \mathbb{Q}(i)[z]$

Step 1. Factorization of $p(z)$ as the product of irreducible factors $p(z) = a_0 q_1(z) \cdots q_m(z)$, $0 \neq a_0 \in \mathbb{C}$.

Step 2. $p_1(z) := 1$, $p_2(z) := a_0$. For $j = 1, \ldots, m$, by the Schur rule to check that all roots of $q_j(z)$ lie inside the unit circle. If the result is true, $p_1(z) := p_1(z) \cdot q_j(z)$, else to check that all roots of $q_j(z)$ lie outside the unit circle. If the result is true, $p_2(z) := p_2(z) \cdot q_j(z)$, else the exact factorization does not exist.

Step 3. Output: $p_-(z) := z^{-\varkappa} p_1(z)$, $\varkappa := \deg p_1(z)$, $p_+(z) := p_2(z)$ or "the exact factorization does not exist".

Theorem

$a(z) \in \mathbb{Q}(i)^{p\times p}[z]$ *is exactly factorable* $\Leftrightarrow$ $\det a(z)$ *is exactly factorable.*

Algorithm 2. The exact factorization of matrix polynomials.
Initial data. $d^{\varkappa}_{-\varkappa}$, $d_j = \frac{1}{2\pi i}\int_{|t|=1} t^{-j-1}\Delta_-^{-1}(z)a(z)\,dt$,
$\det a(z) = \Delta_-(z)z^{\varkappa}\Delta_+(z)$.

- Step 1. Construct the exact factorization of $\det a(z)$.
- Step 2. Find the finite number of the Laurent coefficients of $\Delta_-^{-1}(z)a(z)$.
- Step 3. Find the ranks and the null spaces for matrices T_k.
- Step 4. Find the indices and the FEP of $d^{\varkappa}_{-\varkappa}$.
- Step 5. Construct the exact factorization of $a(z)$.

Remark

All steps are realized in the exact arithmetic.

The following solvers are designed:

- Algorithm 1 was implemented in SolverExactPF in Maple and in solverexactpf in SymPy.
- Algorithm 2 was implemented in SolverExactMPF in Maple and in solverexactmpf in SymPy.
- An algorithm of numerical factorization of a scalar polynomial was implemented in SolverPF in Maple.

Remark

A detailed description of Algorithm 1-2 is presented in the paper "An algorithm for the exact Wiener–Hopf factorization of matrix polynomials and its implementation" prepared for printing.
The algorithm of numerical factorization of a scalar polynomial is described in "Algorithm of polynomial factorization and its implementation in Maple", Bulletin of SUSU, "Mathematical nodelling, programming and computer software", 2018, vol. 11, no.4, pp.110–122.

The nearest goal is to design a solver for numerical solving of the factorization problem for matrix polynomials. Numerical experiments with SolverMPF give hope that this is possible.

The following goals are solvers for other classes.

Thank you for your attention!