Fast Decoding of Codes in the Rank, Subspace, and Sum-Rank Metric

Hannes Bartz, *Member, IEEE,* Thomas Jerkovits, *Member, IEEE,* Sven Puchinger, *Member, IEEE,* Johan Rosenkilde

Abstract

We speed up existing decoding algorithms for three code classes in different metrics: interleaved Gabidulin codes in the rank metric, lifted interleaved Gabidulin codes in the subspace metric, and linearized Reed–Solomon codes in the sum-rank metric. The speed-ups are achieved by reducing the core of the underlying computational problems of the decoders to one common tool: computing left and right approximant bases of matrices over skew polynomial rings. To accomplish this, we describe a skew-analogue of the existing PM-Basis algorithm for matrices over usual polynomials. This captures the bulk of the work in multiplication of skew polynomials, and the complexity benefit comes from existing algorithms performing this faster than in classical quadratic complexity. The new faster algorithms for the various decoding-related computational problems are interesting in their own and have further applications, in particular parts of decoders of several other codes and foundational problems related to the remainder-evaluation of skew polynomials.

Index Terms

Rank Metric, Subspace Metric, Sum-Rank Metric, Interleaved Gabidulin Codes, Lifted Interleaved Gabidulin Codes, Linearized Reed–Solomon Codes, Fast Decoding, (Minimal) Approximant Basis, Interpolation-Based Decoding

I. Introduction

We consider algorithms for decoding certain codes in three different metrics – rank, subspace and sum-rank metric – all of which arise as evaluation-like codes of skew polynomials. Skew polynomials are non-commutative polynomials, where the right multiplication of a scalar $\alpha \in \mathbb{F}_{q^m}$ and the indeterminate x is given as $x\alpha = \sigma(\alpha)x$, where σ is an automorphism of $\mathbb{F}_{q^m}$. The ring of these polynomials is denoted $\mathbb{F}_{q^m}[x;\sigma]$; see Section II-B for the formal definition.

We consider existing decoding principles for the codes and show for each how to speed it up by reducing the core computation to an *approximant basis* computation of matrices over the relevant skew polynomial ring. A reduction to a similar problem for matrices over ordinary polynomial rings has proved beneficial in speeding up decoding of a number of evaluation codes in the Hamming metric and its soft relaxations [2]. Given a matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ and an "order" $d \in \mathbb{Z}_{\geq 0}$, a left approximant basis is a matrix $\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times a}$ such that $\boldsymbol{BA} \equiv 0 \bmod_{\mathrm{r}} x^d$, and such that $\boldsymbol{B}$ is in a certain normal form while satisfying that any vector $\boldsymbol{b} \in \mathbb{F}_{q^m}[x;\sigma]^{1\times a}$ such that $\boldsymbol{bA} \equiv 0 \bmod_{\mathrm{r}} x^d$ is in the left $\mathbb{F}_{q^m}[x;\sigma]$-row space of $\boldsymbol{A}$, see Section III. An analogous definition is given for right approximant bases. Approximant bases for skew polynomials (more generally, for Ore polynomials) were introduced in [3] (under the name "order basis").

A. Main Results

Our central computational result (Theorem 11) is an algorithm for computing a right or left minimal approximant basis of an $a \times b$ matrix of order d, whose complexity's dependency on the order d is only $\mathcal{M}_{q,m}(d)$ (see Table II for more details), where $\mathcal{M}_{q,m}(d)$ is the cost of multiplying two skew polynomials of degree at most d (see Section II-A). The algorithm is a right resp. left adaption of the PM-Basis algorithm for computing minimal approximants over ordinary polynomial rings [11].

In Sections IV and V, we provide new speed records for decoding certain codes in the rank, subspace, and sum-rank metric; see Table I on the following page for a summary.

Each of these speed records are achieved by replacing the bottleneck computations in an existing decoding principle with a left or right minimal approximant basis. To enable these results, we give fast algorithms for a number of decoding-related computational problems which we believe may be interesting in their own right. Most of these new algorithms rely on fast computation of approximant bases. See Table II on the next page for an overview of these problems.

Parts of this paper have been presented at the *2018 IEEE Information Theory Workshop (ITW)* [1].

H. Bartz and T. Jerkovits are with the Institute of Communications and Navigation, German Aerospace Center (DLR), Germany (e-mail: {hannes.bartz, thomas.jerkovits}@dlr.de). T. Jerkovits is also with the Institute for Communications Engineering, Technical University of Munich (TUM), Germany.

S. Puchinger and J. Rosenkilde are with the Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Denmark (e-mail: {svepu, jsrn}@dtu.dk).

Sven Puchinger has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement no. 713683. This work was partly done while Sven Puchinger was at Technical University of Munich, where he was supported by the German Israeli Project Cooperation (DIP) grant no. KR3517/9-1.

TABLE I

OVERVIEW OF NEW DECODING SPEEDS. PARAMETERS: CODE LENGTH n, INTERLEAVING PARAMETER ℓ (USUALLY $\ell \ll n$). FOR SUBSPACE CODES, n_t RESP. n_r IS THE DIMENSION OF THE TRANSMITTED RESP. RECEIVED SUBSPACE. $\mathcal{M}_{q,m}(n)$ IS THE COST OF MULTIPLYING TWO SKEW-POLYNOMIALS OF DEGREE AT MOST n AND ω IS THE MATRIX MULTIPLICATION EXPONENT, SEE SECTIONS II-A AND II-D.

Metric	Code Class	Previously-Fastest Decoder (over $\mathbb{F}_{q^m}$)	Considered Decoder & Complexity (over $\mathbb{F}_{q^m}$)	Our Complexity	Reference
Rank	Interleaved Gabidulin	$\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$ [4]	$O(\ell^2 n^2)$ [5]	$\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$	Theorem 15 Section IV
Subspace	Lifted Interleaved Gabidulin	$O(\ell^2 \max\{n_t, n_r\}^2)$ [6]	see previously fastest	$\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(\max\{n_t, n_r\}))$ plus $O(\ell m n_r^{\omega-1})$ operations in $\mathbb{F}_q$	Theorem 16 Section IV
Sum-Rank/ Skew	Linearized/Skew Reed–Solomon	$O(n^2)$ [7]	see previously fastest	$\tilde{O}(\mathcal{M}_{q,m}(n))$	Theorem 29 Section V

TABLE II

OVERVIEW OF COMPUTATIONAL TOOLS USED TO ACHIEVE FASTER DECODING ALGORITHMS WITH THE COMPLEXITY OF EXISTING ALGORITHMS AND THE PROPOSED ONES. WE INDICATE THE METRIC WHICH THE COMPUTATIONAL PROBLEM IS A PRIORI RELEVANT FOR (R=RANK, S=SUBSPACE, AND SR=SUM-RANK METRIC), AND INDICATE OTHER POTENTIAL APPLICATIONS DISCUSSED IN SECTION VI-B. FOR $\mathcal{M}_{q,m}(n)$ AND ω, SEE TABLE I ABOVE.

Computational Problem	Previous Complexity ($\mathbb{F}_{q^m}$)	Our Complexity	R	S	Sr	Further Applications
Computation of a right/left $\boldsymbol{s}$-ordered weak-Popov approximant basis of order d of an $a \times b$ skew-polynomial matrix (Definition 5)	$O(a^3 b^2 d^2)$ [3] (left case only)	Left/right case, respectively: $\tilde{O}\big(a^{\omega-1} \max\{a,b\} \mathcal{M}_{q,m}(d)\big)$, $\tilde{O}\big(\max\{a,b\} b^{\omega-1} \mathcal{M}_{q,m}(d)\big)$ (Theorem 11 in Section III-C)	X	X	X	
Vector Operator Interpolation (Problem 13) with n interpolation points (vectors in $\mathbb{F}_{q^m}^{\ell+1}$) and degree constraint D (complexities given for $D \in \Theta(n)$).	$O(\ell^2 n^2)$ [8], $\tilde{O}(\ell^3 \mathcal{M}_{q,m}(\ell n))$ on special input [9]	$\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$ plus, under some conditions, $O(\ell m n^{\omega-1})$ operations in $\mathbb{F}_q$ (Theorem 22 in Section IV-B)	X	X		Interpolation step of decoding Mahdavifar–Vardy and (lifted) folded Gabidulin.
Vector Root Finding (Problem 14) for a set of $\ell' \leq \ell+1$ skew polynomial vectors of dimension $\ell+1$, degree at most n, with degree constraints $k^{(1)}, \dots, k^{(\ell)}$ (complexities given for $\max_i k^{(i)} \in \Theta(n)$)	$O(\ell^3 n^2)$ [5], $O(\ell^2 n^2)$ on special input [6]	$\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$ (Theorem 25 in Section IV-C)	X	X		
Remainder-Evaluation Operations (Problem 26): annihilator polynomial computation, multi-point evaluation, and interpolation (number of points and polynomial degrees $\leq n$) of skew polynomials w.r.t. the remainder evaluation.	$O(n^2)$ [7]	$\tilde{O}(\mathcal{M}_{q,m}(n))$ (Theorems 31–32, Section V-B)			X	Encoding linearized/skew Reed–Solomon codes. Repair in the locally repairable / PMDS codes in [10].
2D Vector Remainder Interpolation (Problem 27) with n interpolation points (vectors in $\mathbb{F}_{q^m}^2$).	$O(n^2)$ [7]	$\tilde{O}(\mathcal{M}_{q,m}(n))$ (Theorem 35 in Section V-C)			X	

B. The Studied Codes and Their History

Rank-metric codes are sets of matrices whose distance is measured by the rank of their difference. These codes and their most famous subclass, Gabidulin codes, were independently introduced in [12], [13], [14]. By now, applications of rank-metric codes abound and include criss-cross error correction in memory chips, space-time codes for MIMO systems, code-based cryptography, network coding, distributed data storage, and digital watermarking.

Interleaved Gabidulin codes are direct sums of ℓ Gabidulin codes of the same length over an extension field $\mathbb{F}_{q^m}$: codewords can be represented as an $\mathbb{F}_q^{\ell m \times n}$ matrix by stacking Gabidulin codewords as $\mathbb{F}_q^{m \times n}$ matrices. If such a matrix is subjected to a random error with a low-rank $\mathbb{F}_q$-row dimension, we can correct that error with high probability even if the rank exceeds half the minimum distance of the constituent Gabidulin code. The downside is the rectangular shape of the codewords (since $n \leq m$). Besides being suitable for any application of rank-metric codes with such a rectangular codeword shape, interleaved Gabidulin codes have been explicitly used in works on network coding [15], [16] and code-based cryptography [17], [18].

There are several known polynomial-time decoding algorithms for ℓ-interleaved Gabidulin codes of length n. All of these algorithms correct up to $\frac{\ell}{\ell+1}(n-\overline{k})$ errors, where $\overline{k} := \frac{1}{\ell}\sum_i k_i$ is the mean of the dimensions k_i of the constituent Gabidulin codes. The first-known decoder is due to Loidreau and Overbeck [19]. It is a *partial unique decoder*, which means that for error weights beyond half the minimum distance, it either returns a unique decoding result or fails. The algorithm is based on solving a linear system of equations and has complexity $O(\ell n^\omega)$. Loidreau and Overbeck also derived an upper bound on the relative number of errors of rank t for which the decoder fails. For $t \geq \ell$, it decays exponentially in $m(t-\ell)$. Sidorenko and Bossert [16] proposed a partial unique decoder for interleaved Gabidulin codes that solves a syndrome key equation. The algorithm can be implemented in $O(\ell n^2)$ operations over $\mathbb{F}_{q^m}$ using a Berlekamp–Massey-like algorithm [20] or the demand-driven row reduction algorithm in [21]. There is also a divide-&-conquer approach [4] that solves the key equation in $\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$.

In this paper, we consider the interpolation-based decoder by Wachter-Zeh and Zeh [5], which returns a list of all codewords within a decoding radius at most $\frac{\ell}{\ell+1}(n-\overline{k})$. It can also be seen as a partial unique decoder by declaring a decoding failure if this list is greater than 1. Such a failure event occurs at most in those cases in which the Loidreau–Overbeck decoder fails (see [5, Lemma 8]). The algorithm consists of an *interpolation step* and a *root-finding step* and has complexity $O(\ell^3 n^2)$ operations in $\mathbb{F}_{q^m}$. If there is a unique solution to the decoding problem, then the complexity can be reduced to $O(\ell^2 n^2)$ [6].

Subspace codes are sets of subspaces of a given vector space that have distance properties w.r.t. the *subspace metric* [22]. Beside the initial application of subspace codes as linear authentication codes [23], subspace codes were proposed by Kötter and Kschischang for error correction in network coding [22]. In (random) linear network coding, errors in the network may propagate through the network due to the linear combination of the incoming packets at intermediate nodes. In particular, a single corrupted packet would in turn corrupt all later linear combinations which include this packet. The main idea for subspace codes comes from the observation that the row space of transmitted packets is preserved by the in-network linear operations, and few errors in the network result in a small subspace distance between transmitted and received subspace. Besides the initial constructions of subspace codes based on Gabidulin codes, so called *lifted Gabidulin* codes, in [22], [15], [24], variants with improved error-correction capabilities, including *interleaved* lifted Gabidulin codes [6], were proposed. The currently fasted decoding algorithms for lifted interleaved Gabidulin codes that attain the best decoding region are the syndrome-based approach from [25] which requires $O\big(\ell^3 {n_t}^3\big)$ operations in $\mathbb{F}_{q^m}$ and the interpolation-based decoder from [6] which requires $O\big(\ell^2 \max\{n_t, n_r\}^2\big)$ operations in $\mathbb{F}_{q^m}$, where ℓ is the interleaving order and n_t and n_r are the dimension of the received and transmitted space, respectively. We improve the cost of the latter algorithm.

The sum-rank metric is a family of metrics interpolating the Hamming and rank metric which was first introduced in [26] as being suitable for multi-shot network coding. There are several known codes designed for this metric: partial unit memory codes constructed from rank-metric codes [27], [28], [29], convolutional codes [30], [31], as well as linearized Reed–Solomon codes [32]. The latter codes can be seen as a combination of Reed–Solomon and Gabidulin codes, attain the Singleton bound in the sum-rank metric with equality, and are closely related to skew Reed–Solomon codes in the skew metric [33], [32].

Linearized Reed–Solomon codes have recently shown to provide reliable and secure coding schemes for multi-shot network coding [7]. Furthermore, there is a construction [10] of locally repairable codes with maximal recoverability (also known as partial MDS codes) based on linearized Reed–Solomon codes, which attains the smallest-known field size among all existing code constructions for a wide range of code parameters.

We are aware of two decoding algorithms for linearized and skew Reed–Solomon codes in the literature, both of which are variants of the Welch–Berlekamp decoder for Gabidulin codes [34]. One is due to Boucher [35] and has cubic complexity $O(n^3)$ over $\mathbb{F}_{q^m}$ in the code length n. The other one is quadratic $O(n^2)$ over $\mathbb{F}_{q^m}$ and was presented by Martínez-Peñas and Kschischang [7]. Our work is based on the latter.

C. History of Computational Tools

The history of approximant bases starts with matrices over ordinary polynomials $\mathbb{K}[x]$, for a field $\mathbb{K}$. They are also known as "minimal approximant bases", "order bases", and "σ-bases", and arose as matrix generalizations of simultaneous and Hermite Padé approximations through a range of papers in the 1990's, especially [36], [37], [38]; the latter paper presents fairly efficient algorithms for computing approximant bases. "Shifted" approximant bases were also introduced in these papers. An immediate application of an approximant basis of a matrix $\boldsymbol{F}$ is that a subset of its rows form a generating set for all small-degree vectors in the (left resp. right) kernel of $\boldsymbol{F}$. Several other computations on polynomial matrices can be reduced to approximant bases, e.g. row reduced forms [11], [39]; determinants [11]; Popov and Hermite form [40]; even more general approximations [2]; full-rank bases and unimodular completion [41]; and kernel bases [42]. Computing a (left) approximant basis of $\boldsymbol{A} \in \mathbb{K}[x]^{a\times b}$ with $a \leq b$ in roughly the time it takes to multiply two $a \times a$ polynomial matrices together was given as the PM-Basis algorithm in [11]. For $a \gg b$, this cost can be improved using "partial linearization", see [43] for unshifted or slightly shifted matrices, and [44], [45] for the general case which requires many more tools.

The notion of approximant is based on "row reducedness", see e.g. [46], which is a $\mathbb{K}[x]$-polynomial matrix whose rows have degree among matrices whose rows span the same $\mathbb{K}[x]$-module. The Popov form is a row reduced form that is normalised to be canonical [47], and the weak Popov form is in between these [48]. It seems computationally somewhat more challenging to efficiently compute a reduced form of a matrix than to compute an approximant basis, and the fastest techniques we currently know in the commutative case effectively reduce the former to the latter [11], [40]. Many problems in coding theory which can be solved by approximant bases can instead be solved by row reduction, see e.g. [49].

Turning to the non-commutative case, then approximant bases for matrices over skew polynomials, or more generally Ore polynomials (see Section II-B), were introduced in [3]. That paper, as well as much other literature on computations on Ore polynomials, is concerned with the case where $\mathbb{K}$ is infinite so coefficient growth quickly becomes the computational bottleneck. To address this, the algorithm of [3] generalises "fraction-free" techniques from the commutative case [50]. When $\mathbb{K}$ is finite, this is however slower than the algorithms of [38], [11]. In this paper we consider $\mathbb{K} = \mathbb{F}_{q^m}$ and in particular generalise the algorithm of [11]. This turns out to be conceptually straightforward but rather technical. We will also introduce both a left and a right version of the algorithm; the two cases are of course very similar but subtly different.

Row reducedness and Popov forms were introduced for skew polynomial matrices in [3] using a fraction-free approach. In [21], [51] some of us were involved in generalizing the methods of [48], [9] which are more efficient when $\mathbb{K} = \mathbb{F}_{q^m}$, and applied this to some of the same decoding problems that we address in the present paper; the algorithms of the present paper are all asymptotically more efficient, see Table I.

Besides approximant bases, we study several computational problems that are related to the considered decoding algorithms (see Table II).

The interpolation and root-finding steps of the interpolation-based decoders in [5], [6] are instances of the following two computational problems (see Section IV-A): 1) the *vector interpolation problem* (Problem 13) was first considered in [8] to decode Gabidulin, lifted Gabidulin, and Mahdavifar–Vardy codes. The relation to decoding interleaved Gabidulin codes was given in [5] and lifted interleaved Gabidulin codes in [6]. The problem is also called *bivariate interpolation* since its solutions can be seen as formal bivariate polynomials of bounded y-degree with skew-polynomial coefficients, and the problem statement requires these polynomials to satisfy an evaluation condition and degree bound. Hence, it can be seen as the skew-polynomial analog of the Sudan decoder interpolation step. 2) the *vector root-finding problem* (Problem 14) was first considered in [5] for decoding interleaved Gabidulin codes, where also the currently fastest algorithm was given. The problem was also studied in [6] for decoding lifted interleaved Gabidulin codes. The authors of [6] also present an algorithm that is faster if the solution space has cardinality 1.

The two core computational problems of algorithm in [7] for decoding linearized (or skew) Reed–Solomon codes in the sum-rank (or skew) metric are: 1) fast operations with skew polynomials w.r.t. to the remainder evaluation. This type of evaluation was first studied in [33] and the currently fastest algorithms to compute the relevant operations were given in [7]. 2) a 2-dimensional vector remainder interpolation, which can be seen as the analog of the Welch–Berlekamp reconstruction problem for skew polynomials w.r.t. the remainder evaluation. This problem was first studied in [52], and later in [35], [7]. The currently fastest algorithm to solve this problem was proposed in [7].

D. Reader's Guide

We set notation, define our cost model, and recall known results on skew polynomials in Section II. In Section III, we analyze left and right approximant bases over skew polynomial rings and propose new, faster, algorithms to compute them. These results lay the foundation for the remainder of the paper, which discusses computational problems related to decoding rank-metric and subspace codes (Section IV) as well as sum-rank-metric codes (Section IV). These two sections are independent of each other. Both of them start by a subsection that formally states the relevant computational problems (cf. Table II) and recalls their relation to the considered decoders. The respective remaining subsections propose new algorithms to solve these computational problems. We conclude the paper in Section VI, including several remarks on generality, further applications of the results, and some open problems. The appendix includes some extended results out of the main scope of the paper, as well as examples.

II. Preliminaries

Let q be a prime power, m be a positive integer, and denote by $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$ the finite field of size q and q^m, respectively. The field $\mathbb{F}_{q^m}$ is an extension field of $\mathbb{F}_q$ of extension degree m and hence also a vector space over $\mathbb{F}_q$ of dimension m. The Galois group of the extension is cyclic and consists of the powers of the Frobenius automorphism $\cdot^q : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, $\alpha \mapsto \alpha^q$, i.e., $\mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q) = \{\cdot^{q^i} : i = 0, \dots, m-1\}$. The generators of the Galois group are the $\cdot^{q^i}$ with $\gcd(i, m) = 1$.

A. Cost Model

We use the big-O notation family to state asymptotic costs of algorithms, and $\tilde{O}(\cdot)$ which neglects logarithmic terms in the input parameter. Furthermore, we express the cost of algorithms either in arithmetic operations over the field $\mathbb{F}_{q^m}$ or over $\mathbb{F}_q$: here we include not only $+, -, \cdot$ and $/$, but also applications of a (specific) automorphism $\sigma \in \mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$. This is uncommon in the literature on computation at large, but has become standard for work on Gabidulin codes and related codes. The basic reasoning is that if the extension $\mathbb{F}_{q^m} : \mathbb{F}_q$ is built using a normal basis (see, e.g., [53]) and $\sigma = (\cdot)^{q^i}$, then $\sigma(a)$ is simply the cyclic shift of i positions of the vector description of a over $\mathbb{F}_q$ in that basis. However, multiplication is not a priori as efficient in normal bases as it is in power bases, and the complications arise when attempting requiring that all operations are fast simultaneously. We let $\mathcal{F}(m)$ denote an upper bound on the cost of all of these operations in $\mathbb{F}_{q^m}$ in terms of operations in $\mathbb{F}_q$. Couveignes and Lercier [54] showed that it is possible to choose a basis such that $\mathcal{F}(m) \in \tilde{O}(m)$, and we will mostly assume such a basis. In practice and for small m it might well be faster to use either a power bases with $\mathcal{F}(m) \in \tilde{O}(m^2)$ (bottleneck being applications of σ) or a normal basis with $\mathcal{F}(m) \in O(m^2)$ (bottleneck being multiplication and division).

In cost bounds, we denote by ω the matrix multiplication exponent, i.e. the infimum of values $\omega_0 \in [2; 3]$ such that there is an algorithm for multiplying $n \times n$ matrices over $\mathbb{F}_{q^m}$ in time $O(n^{\omega_0})$ for $n \to \infty$. The currently known best bound is $\omega < 2.37286$ [55].

B. Skew Polynomials

In this paper, all codes and algorithms are defined over skew polynomials which are non-commutative polynomials and were introduced by Ore in [56]; for this reason they are also known as Ore polynomial rings. The general construction over any field $\mathbb{K}$ uses an endomorphism σ and a "σ-derivation" $\delta : \mathbb{K} \to \mathbb{K}$, and can be used for unifying theoretical and computational questions on linear differential equations, time-dependent systems and recursively defined sequences of numbers, see e.g. [57], sometimes in the specialisation of D-finiteness, see e.g. [58].

We will only use the specialisation where $\mathbb{K} = \mathbb{F}_{q^m}$, $\sigma = (\cdot)^{q^i}$ with $\gcd(i, m) = 1$ (i.e. $\mathrm{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q) = \langle\sigma\rangle$), and $\delta = 0$. When $i = 1$, these rings are isomorphic to linearized polynomials, which were also introduced by Ore [59], and for $i > 1$ behave in much the same way. Besides their applications in coding theory, these are studied in cryptography [17], dynamical systems [60], and are of theoretical interest [59], [61], [62]. In the remainder of the paper, when we say "skew polynomials", we mean this restricted setting. They are sometimes also called *twisted polynomials* or *σ-polynomials*.

A *skew polynomial* (in our restricted setting) is then a formal polynomial sum $f = \sum_{i\geq 0} f_i x^i$, indexed by powers of an indeterminant x, and with only a finite number of $f_i \in \mathbb{F}_{q^m}$ being non-zero. We add two polynomials monomial-wise as for usual polynomials. Multiplication of skew polynomials is defined by the rule

$$x \cdot a = \sigma(a) \cdot x$$

for any $a \in \mathbb{F}_{q^m}$. By associativity and distributivity, we have

$$f \cdot g = \sum_{i\geq 0} \Big(\sum_{j\geq 0} f_j \sigma^j(g_{i-j}) \Big) x^i. \tag{1}$$

for any two skew polynomials $f = \sum_i f_i x^i$ and $g = \sum_j g_j x^j$, where we define the $f_i = g_i = 0$ for $i < 0$. The set of skew polynomials with this addition and multiplication rule is a non-commutative integral domain and denoted by $\mathbb{F}_{q^m}[x; \sigma]$.

The *degree* of a skew polynomial is defined by

$$\deg f := \begin{cases} \max\{i \,:\, f_i \neq 0\}, & \text{if } f \neq 0, \\ -\infty, & \text{otherwise.} \end{cases}$$

As for ordinary polynomials, we have $\deg(f \cdot g) = \deg f + \deg g$, and $\deg(f + g) \leq \max\{\deg f, \deg g\}$, where equality holds in the latter iff $\deg f \neq \deg g$ or $\deg f = \deg g$ and the leading coefficients of f and g do not sum to zero.

There is both a left and right division algorithm, hence the ring is left and right Euclidean. Let $f, g, h \in \mathbb{F}_{q^m}[x; \sigma]$ such that $h \neq 0$. We denote the remainder of the left division of f by h as $f \bmod_{\mathrm{l}} h$, i.e., $f \bmod_{\mathrm{l}} h$ is the unique skew polynomial of degree $< \deg h$ for which $f \bmod_{\mathrm{l}} h = f - h\chi$ for some $\chi \in \mathbb{F}_{q^m}[x; \sigma]$. Analogously, the remainder w.r.t. the right division is denoted by $f \bmod_{\mathrm{r}} h$ (in this case we have $f \bmod_{\mathrm{r}} h = f - h\chi$ for some $\chi \in \mathbb{F}_{q^m}[x; \sigma]$). We say that f and g are congruent left-modulo h, written $f \equiv g \bmod_{\mathrm{l}} h$, if $f - g$ is divisible by h from the left (i.e., $(f - g) \bmod_{\mathrm{l}} h = 0$). Likewise, $f \equiv g \bmod_{\mathrm{r}} h$ if $(f - g) \bmod_{\mathrm{r}} h = 0$.

Since $\mathbb{F}_{q^m}[x; \sigma]$ is left and right Euclidean, it is also a left and right principal ideal domain. This implies that left and right modules over $\mathbb{F}_{q^m}[x; \sigma]$ share many important properties with modules over $\mathbb{F}_{q^m}[x]$. For instance, any left or right submodule of $\mathbb{F}_{q^m}[x; \sigma]^a$ is free and any two basis of such a submodule have the same number of elements. Hence, the rank of a module is well-defined. Furthermore, two $a \times b$ matrices $\boldsymbol{B}_1, \boldsymbol{B}_2$ over $\mathbb{F}_{q^m}[x; \sigma]$ generate the same left row (or right column) space if and only if there is an invertible $a \times a$ ($b \times b$) matrix $\boldsymbol{U}$ with $\boldsymbol{B}_1 = \boldsymbol{U}\boldsymbol{B}_2$ ($\boldsymbol{B}_1 = \boldsymbol{B}_2\boldsymbol{U}$, resp.). See, e.g., [63] for more details.

C. Evaluations of Skew Polynomials

It turns out that skew polynomials give rise to multiple notions of mappings [33] which behave similarly to evaluation of ordinary polynomials, and these can each be used to build "evaluation codes" from skew polynomials. In this paper, we consider two such "evaluations":

- operator evaluation (used in Section IV) and
- remainder evaluation (used in Section V).

We will distinguish the two evaluation types notationally by their brackets (soft for operator and square for remainder evaluation), see below.

The *operator evaluation map* of a skew polynomial $f = \sum_i f_i x^i \in \mathbb{F}_{q^m}[x; \sigma]$ is defined as

$$f(\cdot) \,:\, \mathbb{F}_{q^m} \to \mathbb{F}_{q^m},\ \alpha \mapsto \textstyle\sum_i f_i \sigma^i(\alpha).$$

For any $f, g \in \mathbb{F}_{q^m}[x; \sigma]$ and $\alpha \in \mathbb{F}_{q^m}$, we have the following sum and product rule:

$$\begin{aligned} (f + g)(\alpha) &= f(\alpha) + g(\alpha) \\ (f \cdot g)(\alpha) &= f(g(\alpha)). \end{aligned}$$

Since σ is an $\mathbb{F}_q$-linear map, also $f(\cdot)$ is an $\mathbb{F}_q$-linear map and the (operator) root space $\ker f(\cdot) := \{\alpha \in \mathbb{F}_{q^m} \mid f(\alpha) = 0\}$ is an $\mathbb{F}_q$-vector space. Furthermore, we have $\dim \ker f(\cdot) \leq \deg f$ for any non-zero $f \in \mathbb{F}_{q^m}[x;\sigma]$.

For codes we will consider evaluating a skew polynomial f at multiple values $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m}$ which are linearly indpendent over $\mathbb{F}_q$, for which the following constructions of skew polynomials are crucial:

- Let $\mathcal{U} \subseteq \mathbb{F}_{q^m}$ be the $\mathbb{F}_q$-subspace spanned by $\alpha_1, \ldots, \alpha_n$. Then there is a unique monic skew polynomial $\mathcal{M}^{\text{op}}_{\mathcal{U}}$, called *(operator) annihilator polynomial of* $\mathcal{U}$ [64], [65] (also called *minimal subspace polynomial*) with $\ker \mathcal{M}^{\text{op}}_{\mathcal{U}}(\cdot) = \mathcal{U}$ and $\deg \mathcal{M}^{\text{op}}_{\mathcal{U}} = \dim_{\mathbb{F}_q}(\mathcal{U}) = n$.
- If $\mathbb{F}_{q^m}[x;\sigma]_{<n}$ denotes all skew polynomials of degree less than n, then $\mathbb{F}_{q^m}[x;\sigma]_{<n}$ is in bijection with $\mathbb{F}_{q^m}^n$ through operator evaluation at $\alpha_1, \ldots, \alpha_n$. In other words, for any $r_1, \ldots, r_n \in \mathbb{F}_{q^m}$, there is a unique skew polynomial $\mathcal{I}^{\text{op}}_{\{(\alpha_i, r_i)\}_{i=1}^n}$, called the *(operator) interpolation polynomial*, of degree $< n$ such that $\mathcal{I}^{\text{op}}_{\{(\alpha_i, r_i)\}_{i=1}^n}(\alpha_i) = r_i$ for all $i = 1, \ldots, n$ [66], [65].

The *remainder evaluation map* of a skew polynomial $f \in \mathbb{F}_{q^m}[x;\sigma]$ is defined by

$$f[\cdot] : \mathbb{F}_{q^m} \mapsto \mathbb{F}_{q^m}, \quad \alpha \mapsto f \bmod_{\text{r}} (x - \alpha).$$

For any $f, g \in \mathbb{F}_{q^m}[x;\sigma]$ and $\alpha \in \mathbb{F}_{q^m}$, we have [67]

$$(f+g)[\alpha] = f[\alpha] + g[\alpha]$$
$$(f \cdot g)[\alpha] = \begin{cases} 0, & \text{if } c = 0, \\ f\left[\frac{\sigma(c)\alpha}{c}\right] c, & \text{if } c \neq 0, \end{cases}$$

where $c := g[\alpha]$. There are analogs of annihilator and interpolation polynomials for the remainder evaluation. However, since their definition requires further notation and is only relevant in Section V, we will discuss these notions at the start of that section.

For more details on the evaluation maps and their differences, we refer to [33]. Throughout the paper, whenever it is clear from the context which evaluation map we mean, we omit the prefixes "operator" and "remainder".

D. Cost of Operations with Skew Polynomials

We denote by $\mathcal{M}_{q,m}(n)$ the cost of multiplying two skew polynomials over $\mathbb{F}_{q^m}/\mathbb{F}_q$ of degree n. The best-known cost bounds on $\mathcal{M}_{q,m}(n)$ are

$$\mathcal{M}_{q,m}(n) \in \tilde{O}\big(\min\{n^{\omega-2}m^2,\, nm^{\omega-1}\}\big)$$

operations over $\mathbb{F}_q$ using the algorithms in [68], [69] and

$$\mathcal{M}_{q,m}(n) \in O\Big(n^{\min\{\frac{\omega+1}{2}, 1.635\}}\Big)$$

operations over $\mathbb{F}_{q^m}$ using the algorithm in [70]. Using a basis with $\mathcal{F}(m) \in \tilde{O}(m)$, and assuming $(\omega+1)/2 > 1.635$, i.e. $\omega > 2.27$, the algorithms in [68], [69] provide the best cost bounds whenever $n \in \Omega\big(m^{\frac{2}{5-\omega}}\big)$, while [70] provides the best cost bound when $n \in O\big(m^{\frac{2}{5-\omega}}\big)$.

All algorithms are faster than classical multiplication which has quadratic complexity $\Theta(n^2)$ operations over $\mathbb{F}_{q^m}$. This is obvious for the multiplication algorithm in [70] (exponent is reduced from 2 to ≤ 1.635), and holds for the one in [68] due to

$$\tilde{O}\big(\min\{n^{\omega-2}m^2,\, nm^{\omega-1}\}\big) \subseteq o\big(n^2\mathcal{F}(m)\big)\,.$$

By combining the results in [69], [68], [70], [71], the following skew polynomial operations can be performed in $\tilde{O}(\mathcal{M}_{q,m}(n))$ time:

- Left and right division of two skew polynomials of degree at most n.
- Operator evaluation of a skew polynomial of degree $\leq n$ at n field elements (*multi-point (operator) evaluation*).
- Compute the operator annihilator polynomial $\mathcal{M}^{\text{op}}_{\mathcal{U}}$ of an n-dimensional subspace $\mathcal{U}$.
- Compute an operator interpolation polynomial at n field elements.

In Section V, we will discuss the remainder-evaluation analogs of the latter three operations. We did not find the analog of the above computational cost bounds extant in the literature, so we show in Section V-B that they can also be performed in $\tilde{O}(\mathcal{M}_{q,m}(n))$ time.

III. Approximant Bases Over $\mathbb{F}_{q^m}[x;\sigma]$

In this section, we study the central computational object that will enable us to speed up decoding algorithms and computational tools discussed in later sections: approximant bases over skew polynomial rings. Here, we use the notation and adapt the algorithms of [72], which studied these bases over ordinary polynomial rings. For skew polynomials over finite fields, the resulting algorithms have smaller complexity than the previously fastest method in [3].

A. Modules and Matrices over Skew Polynomial Rings

For a matrix $\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ and $\boldsymbol{s} \in \mathbb{Z}^a$, we define the $\boldsymbol{s}$-shifted column degree of $\boldsymbol{B}$ to be the tuple

$$\operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}) = [d_1, \dots, d_b] \in (\mathbb{Z} \cup \{-\infty\})^b$$

where d_j is the maximal shifted degree in the j-th column, i.e., $d_j := \max_{i=1,\dots,a}\{\deg B_{ij} + s_i\}$. We write $\operatorname{cdeg}(\boldsymbol{B}) := \operatorname{cdeg}_{\boldsymbol{0}}(\boldsymbol{B})$, where $\boldsymbol{0} := [0,\dots,0]$. Analogously, for $\boldsymbol{s} \in \mathbb{Z}^b$, we define the ($\boldsymbol{s}$-shifted) row degree of $\boldsymbol{B}$ to be

$$\operatorname{rdeg}_{\boldsymbol{s}}\boldsymbol{B} := \operatorname{cdeg}_{\boldsymbol{s}}\left(\boldsymbol{B}^\top\right) \quad \text{and} \quad \operatorname{rdeg}\boldsymbol{B} := \operatorname{cdeg}\left(\boldsymbol{B}^\top\right).$$

The degree of the matrix, i.e. the maximal degree among its entries, is denoted:

$$\deg \boldsymbol{B} := \max_{i,j}\{\deg B_{ij}\}.$$

If $\boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^{1\times a} \setminus \{\boldsymbol{0}\}$ is a row vector and $\boldsymbol{s} = [s_1,\dots,s_a] \in \mathbb{Z}^a$ a shift, we define the $\boldsymbol{s}$-pivot index of $\boldsymbol{v}$ to be the largest index i with $1 \le i \le a$ such that $\deg v_i + s_i = \operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{v})$, and analogously for column vectors. If $a \ge b$ (or $a \le b$, respectively), then we say that $\boldsymbol{B}$ is in column (row) $\boldsymbol{s}$-ordered weak Popov form if the $\boldsymbol{s}$-pivot indices of its columns (rows) are distinct and non-decreasing in the column (row) index.

The next two lemmas present key properties of matrices in row or column weak Popov form that we will use later in this section. The first one is a variant of the "predictable degree property", see [46], which is central to row- or column-reduced matrices such as those in ordered row or column weak Popov form. An analogous result holds for singular rank or non-square matrices, but we will need it only for square ones.

Lemma 1. *Let $\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$ be full rank and $\boldsymbol{s} \in \mathbb{Z}^b$.*

- *"Column case": Assume $\boldsymbol{B}$ is in $\boldsymbol{s}$-ordered column weak Popov form, $\boldsymbol{t} := \operatorname{cdeg}_{\boldsymbol{s}} \boldsymbol{B}$, and $\boldsymbol{p} = \boldsymbol{B}\boldsymbol{\lambda}$ for non-zero column vectors $\boldsymbol{p}, \boldsymbol{\lambda} \in \mathbb{F}_{q^m}[x;\sigma]^{b\times 1}$. Then*
  - $\operatorname{cdeg}_{\boldsymbol{s}} \boldsymbol{p} = \operatorname{cdeg}_{\boldsymbol{t}} \boldsymbol{\lambda}$ *and*
  - *the $\boldsymbol{s}$-pivot index of $\boldsymbol{p}$ equals the $\boldsymbol{t}$-pivot index of $\boldsymbol{\lambda}$.*
- *"Row case": Assume $\boldsymbol{B}$ is in $\boldsymbol{s}$-ordered row weak Popov form, $\boldsymbol{t} := \operatorname{rdeg}_{\boldsymbol{s}}\boldsymbol{B}$, and $\boldsymbol{p} = \boldsymbol{\lambda}\boldsymbol{B}$ for non-zero row vectors $\boldsymbol{p}, \boldsymbol{\lambda} \in \mathbb{F}_{q^m}[x;\sigma]^{1\times b}$. Then*
  - $\operatorname{rdeg}_{\boldsymbol{s}}\boldsymbol{p} = \operatorname{rdeg}_{\boldsymbol{t}}\boldsymbol{\lambda}$ *and*
  - *the $\boldsymbol{s}$-pivot index of $\boldsymbol{p}$ equals the $\boldsymbol{t}$-pivot index of λ.*

Proof. We first prove the column case. Let $\mu := \operatorname{cdeg}_{\boldsymbol{t}} \boldsymbol{\lambda}$ and h be the $\boldsymbol{t}$-pivot intex of $\boldsymbol{\lambda}$. Since $\boldsymbol{p} = \boldsymbol{B}\boldsymbol{\lambda}$, then $\deg p_i \le \max_{j=1,\dots,b}\{\deg F_{ij} + \deg \lambda_j\} \le \max_{j=1,\dots,b}\{t_j - s_i + \deg \lambda_j\}$, and so $\operatorname{cdeg}_{\boldsymbol{s}} \boldsymbol{p} \le \mu$. Let $\boldsymbol{u} \in \mathbb{F}_{q^m}^{b\times 1}$ be the vector whose i-th entry is the $x^{[\mu - s_i]}$-coefficient of p_i (the coefficient is zero if $\deg p_i < \mu - s_i$). Hence, $\operatorname{cdeg}_{\boldsymbol{s}} \boldsymbol{p} = \mu$ iff $\boldsymbol{u} \neq \boldsymbol{0}$. Further, if $\boldsymbol{u} \neq \boldsymbol{0}$, then the $\boldsymbol{s}$-pivot index of $\boldsymbol{p}$ is the greatest non-zero index of $\boldsymbol{u}$.

Since $\deg F_{ij} \le t_j - s_i$ and $\deg \lambda_j \le \mu - t_j$, the entries of $\boldsymbol{u}$ only depend on some of the leading coefficients in the matrix $\boldsymbol{B}$ and vector $\boldsymbol{\lambda}$. Let $\operatorname{lm}_{\boldsymbol{s}}(\boldsymbol{B})$ be the $\boldsymbol{s}$-leading matrix of $\boldsymbol{B}$ whose (i,j)-th entry is the $x^{t_j - s_i}$-coefficient of F_{ij}, defined as 0 if $\deg F_{ij} < t_j - s_i$. Similarly, define l_j to be the $x^{[\mu - t_j]}$-coefficient of λ_j. Then, by the definition of linearized polynomial multiplication, u_i is the inner product of the i-th row of $\operatorname{lm}_{\boldsymbol{s}}(\boldsymbol{B})$ and the vector $\boldsymbol{l}_i := [\sigma^{t_1 - s_i}(l_1), \dots, \sigma^{t_b - s_i}(l_b)]^\top$.

Since $\boldsymbol{B}$ is full-rank and in $\boldsymbol{s}$-ordered column weak Popov form, the $\boldsymbol{s}$-pivot index of its j-th column is j and $\operatorname{lm}_{\boldsymbol{s}}(\boldsymbol{B})$ is in upper triangular form with only non-zero entries on its diagonal. Also, $\boldsymbol{l}_i \neq \boldsymbol{0}$ since at least one λ_j fulfills $\deg \lambda_j + t_j = \mu$, and h as defined above is the greatest non-zero index of $\boldsymbol{l}_i$ (independent of i). Thus, u_h is non-zero and h is also the greatest non-zero index of $\boldsymbol{u}$, which proves the claim.

The row case follows analogously, the only differences being that $\operatorname{lm}_{\boldsymbol{s}}(\boldsymbol{B})$ is defined as the matrix containing the $x^{t_i - s_j}$-coefficient of F_{ij} (which is in lower triangular form), and that u_i is the inner product of the vector $\boldsymbol{l}_i := [l_1,\dots,l_b]$ (no automorphisms applied) and the i-th column of $\operatorname{lm}_{\boldsymbol{s}}(\boldsymbol{B})$ with entry-wise some automorphisms applied. This does not change the argument above since automorphisms do not map non-zero entries to zero. □

Remark 2. *The predictable degree property (Lemma 1) was studied for* row-reduced *matrices over skew polynomials in [73, Lemma A.1]. More precisely, the property $\operatorname{rdeg}_{\boldsymbol{s}}\boldsymbol{p} = \operatorname{rdeg}_{\boldsymbol{t}}\boldsymbol{\lambda}$ ("row case") was shown for the shift $\boldsymbol{s} = \boldsymbol{0}$. Since "row reduced" is weaker than "ordered weak Popov", pivots of $\boldsymbol{p}$ and $\boldsymbol{\lambda}$ are not necessarily the same in this case.*

The following lemma is the skew analog of [72, Theorem 1.28, case (iii)]. We state the theorem for column weak Popov form and write the row case in parentheses.

Lemma 3. *Let $\boldsymbol{B}_1 \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$ be in $\boldsymbol{s}$-ordered column (row) weak Popov form and $\boldsymbol{B}_2 \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$ be in $\boldsymbol{t}$-ordered column (row) weak Popov form, where $\boldsymbol{t} := \operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$ ($\boldsymbol{t} := \operatorname{rdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$). Then, $\boldsymbol{B}_1\boldsymbol{B}_2$ ($\boldsymbol{B}_2\boldsymbol{B}_1$) is in $\boldsymbol{s}$-ordered column (row) weak Popov form.*

Proof. We prove the column case, the row case follows analogously. Let $\boldsymbol{u} = [u_1, \ldots, u_b] = \operatorname{cdeg}_{\boldsymbol{t}}(\boldsymbol{B}_2)$. Let $\boldsymbol{h}_i$ be the i-th column of $\boldsymbol{B}_1\boldsymbol{B}_2$. Denote by $B_{2,ij}$ the (i,j)-th entry of $\boldsymbol{B}_2$. By Lemma 1 then $\operatorname{cdeg}_{\boldsymbol{s}} \boldsymbol{h}_j = \max_{i=1,\ldots,b}\{\deg B_{2,ij} + t_i\} = u_j$, and further the $\boldsymbol{s}$-pivot index of $\boldsymbol{h}_j$ is $\max\{i : \deg B_{2,ij} + t_i = u_j\}$ which is exactly the $\boldsymbol{t}$-pivot index of the j-th column of $\boldsymbol{B}_2$. Since these are all in strictly increasing order, so must the $\boldsymbol{s}$-pivots of $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_b$. Hence $\boldsymbol{B}_2\boldsymbol{B}_1$ is in ordered weak Popov form. □

B. Approximant Bases over $\mathbb{F}_{q^m}[x;\sigma]$

Let $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ and $d \in \mathbb{Z}_{\geq 0}$. A right approximant of $\boldsymbol{A}$ of order d is a vector $\boldsymbol{b} \in \mathbb{F}_{q^m}[x;\sigma]^{b\times 1}$ such that

$$\boldsymbol{A}\boldsymbol{b} \equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^d.$$

A left approximant of $\boldsymbol{A}$ of order d is $\boldsymbol{b} \in \mathbb{F}_{q^m}[x;\sigma]^{1\times a}$ with

$$\boldsymbol{b}\boldsymbol{A} \equiv \boldsymbol{0} \bmod_{\mathrm{r}} x^d.$$

Lemma 4. *The set of right (left) approximants of $\boldsymbol{A}$ of order d is a free right (left) $\mathbb{F}_{q^m}[x;\sigma]$-module of rank b (rank a).*

Proof. The set is a subset of $\mathbb{F}_{q^m}[x;\sigma]^{b\times 1}$ ($\mathbb{F}_{q^m}[x;\sigma]^{1\times a}$, respectively) and obviously closed under addition and right (left) multiplication by elements of $\mathbb{F}_{q^m}[x;\sigma]$, hence a free right (left) module. Further, the vector $[0, \ldots, 0, x^d, 0, \ldots, 0]$ of suitable length is clearly a right (left) approximant of $\boldsymbol{A}$ of order d, so the module of all right (left) approximants must contain a module of rank b (rank a), hence must themselves be of rank b (rank a) since it cannot be greater. □

Lemma 4 shows that the following definition is well-posed.

Definition 5 (left/right approximant bases)**.** *Let $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ and $d \in \mathbb{Z}_{\geq 0}$.*

- *For $\boldsymbol{s} \in \mathbb{Z}^b$, a right $\boldsymbol{s}$-ordered weak-Popov approximant basis of $\boldsymbol{A}$ of order d is a full-rank matrix $\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$ s.t.*
  1) *$\boldsymbol{B}$ is in $\boldsymbol{s}$-ordered column weak Popov form.*
  2) *The columns of $\boldsymbol{B}$ are a basis of all right approximants of $\boldsymbol{A}$ of order d.*
- *For $\boldsymbol{s} \in \mathbb{Z}^a$, a left $\boldsymbol{s}$-ordered weak-Popov approximant basis of $\boldsymbol{A}$ of order d is a full-rank matrix $\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times a}$ s.t.*
  1) *$\boldsymbol{B}$ is in $\boldsymbol{s}$-ordered row weak Popov form.*
  2) *The rows of $\boldsymbol{B}$ are a basis of all right approximants of $\boldsymbol{A}$ of order d.*

We denote by $\mathsf{owPopovApprox}_{\mathsf{R}}(\boldsymbol{A}, \boldsymbol{s}, d)$ (right case) and $\mathsf{owPopovApprox}_{\mathsf{L}}(\boldsymbol{A}, \boldsymbol{s}, d)$ (left case) the sets of all such bases, respectively. If the input is not relevant, we simply write (left or right) approximant basis.

Remark 6. *The most common definition in the literature requires approximant bases only to be row-reduced (denoted by "($\boldsymbol{s}$-)minimal approximant basis"). Here, we use a stronger normal form, ordered weak Popov form. The motivation comes from [72], where (over ordinary polynomials) it was shown that the fastest algorithms for computing approximant bases can be adapted to output ordered weak Popov forms at no extra (asymptotic) cost.*

For approximant bases over ordinary polynomial rings, the "row/left" versus the "column/right" view becomes one of notational convenience, since we can trivially obtain one from the other by transposition. In the non-commutative case of approximant bases over skew polynomial rings, this is no longer true (see Example 37 in Appendix B for a counterexample) and the row and column cases are simply slightly different: we need theorems and algorithms tailored to each case, even if most of the statements and proofs are very similar for the two cases.

The currently fastest algorithm to compute a left approximant basis over $\mathbb{F}_{q^m}[x;\sigma]$ (in the weaker "row-reduced form" instead of ordered weak Popov form) is $O(a^3b^2d^2)$ operations in $\mathbb{F}_{q^m}$ [3]. Note that the algorithm in [3] is designed to handle coefficient growth in certain infinite fields, and also the complexity analysis is only done for this case. Our own analysis of the algorithm gives the stated complexity over $\mathbb{F}_{q^m}$.

C. A New Algorithm to Compute Approximant Bases

In this section, we adapt the recursive (left) PM-basis algorithm [74], [72] over ordinary polynomial rings to compute a left and right approximant basis over skew polynomials. For the base case (Section III-C1), we prove that the algorithm over $\mathbb{F}_{q^m}[x]$ can be used with only small modifications. Also the recursion step (Section III-C2) is very similar to the original algorithm, but we need to be careful about the non-commutativity of the skew polynomial ring.

1) Base Case: Right and Left Approximant Bases of Degree 1: In the following, we show how to obtain right and left approximant basis order 1 of a degree 0 matrix. For both sides, we reduce the problem to computing an approximant basis over the ordinary polynomials [74], [72] (cf. Algorithm 1) using suitable bijective mappings between $\mathbb{F}_{q^m}[x]$ and $\mathbb{F}_{q^m}[x;\sigma]$.

For the right case, we use the following mapping φ and its inverse, which we extend to matrices entry-wise.

$$\begin{aligned}\varphi \;:\; \mathbb{F}_{q^m}[x] &\to \mathbb{F}_{q^m}[x;\sigma],\\ \textstyle\sum_i f_i x^i &\mapsto \textstyle\sum_i \sigma^i(f_i)x^i.\end{aligned} \tag{2}$$

We use two important properties of the mapping:

$$\varphi(fh) = \varphi(f)\varphi(h) \quad \forall\, f \in \mathbb{F}_{q^m}[x] \text{ and } h \in \mathbb{F}_{q^m}[x]_{<1}, \tag{3}$$

$$\varphi(fg \bmod x) = \varphi(f)\varphi(g) \bmod_{\mathrm{l}} x \quad \forall\, f,g \in \mathbb{F}_{q^m}[x], \tag{4}$$

where (3) holds due to (denote $h = h_0x^0$)

$$\begin{aligned}\varphi(fh) &= \varphi(\textstyle\sum_i f_i h_0 x^i) = \sum_i \sigma^i(f_i)\sigma^i(h_0)x^i\\ &= (\textstyle\sum_i \sigma^i(f_i)x^i)(h_0x^0) = \varphi(f)\varphi(h)\end{aligned}$$

and (4) is a direct consequence of the first property (write $g = g_0x^0 + (\sum_{i>0} g_i x^i)$ and use the additivity of φ).

The resulting algorithm for computing right skew approximant bases of order 1 is outlined in Algorithm 2. We prove its correctness using the reduction shown in Figure 1.

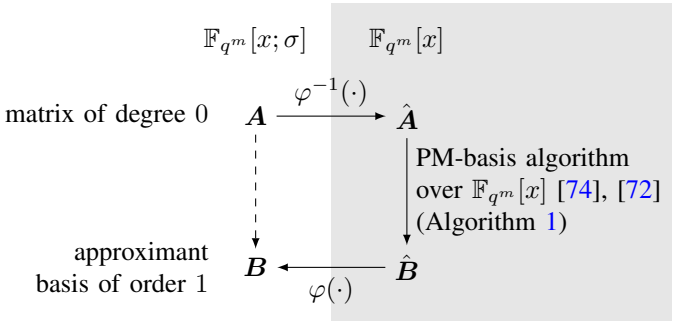

Fig. 1. Illustration of the reduction used in the correctness proof of Algorithm 2 (Theorem 7). Variables are defined as in Algorithm 2.

Algorithm 1: RightBaseCase [74], [72]

Input : matrix $\hat{\boldsymbol{A}} \in \mathbb{F}_{q^m}[x]^{a\times b}$ with $\deg(\hat{\boldsymbol{A}}) < 1$, shifts $\boldsymbol{s} \in \mathbb{Z}^b$
Output: $\hat{\boldsymbol{B}} \in \mathbb{F}_{q^m}[x]^{b\times b}$, a right $\boldsymbol{s}$-ordered weak-Popov approximant basis of $\hat{\boldsymbol{A}}$ of order 1 over $\mathbb{F}_{q^m}[x]$

$\pi_{\boldsymbol{s}} \leftarrow b \times b$ permutation matrix s.t. $[(s_1,1),\dots,(s_b,b)]\pi_{\boldsymbol{s}}$ is lexicographically increasing
$[i_1,\dots,i_\rho],[j_1,\dots,j_\rho] \leftarrow$ row and column rank profiles of $\hat{\boldsymbol{A}}\pi_{\boldsymbol{s}}$ (i.e., the column/row indices of leading ones in a row/column echelon form of $\hat{\boldsymbol{A}}\pi_{\boldsymbol{s}}$) // compute as in [75]
$[k_1,\dots,k_{b-\rho}] \leftarrow \{1,\dots,b\} \setminus \{j_1,\dots,j_\rho\}$ sorted increasingly
$\hat{\boldsymbol{A}}_1 \leftarrow$ submatrix of $\hat{\boldsymbol{A}}\pi_{\boldsymbol{s}}$ with indices in $\{i_1,\dots,i_\rho\} \times \{j_1,\dots,j_\rho\}$
$\hat{\boldsymbol{A}}_2 \leftarrow$ submatrix of $\hat{\boldsymbol{A}}\pi_{\boldsymbol{s}}$ with indices in $\{i_1,\dots,i_\rho\} \times \{k_1,\dots,k_{b-\rho}\}$
$\pi \leftarrow$ permutation s.t. $[j_1 \dots j_\rho k_1 \dots k_{b-\rho}]\pi = [1 \dots b]$
return $\pi_{\boldsymbol{s}}\pi^{-1}\begin{bmatrix} x\boldsymbol{I}_\rho & -\hat{\boldsymbol{A}}_1^{-1}\hat{\boldsymbol{A}}_2\boldsymbol{I}_{b-\rho} \\ \boldsymbol{0} & \boldsymbol{I}_{b-\rho}\end{bmatrix}\pi\pi_{\boldsymbol{s}}^{-1} \in \mathbb{F}_{q^m}[x]^{b\times b}$

Algorithm 2: RightSkewBaseCase

Input : $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ with $\deg(\boldsymbol{A}) < 1$, $\boldsymbol{s} \in \mathbb{Z}^b$
Output: $\boldsymbol{B} \in \mathsf{owPopovApprox}_{\mathsf{R}}(\boldsymbol{A},\boldsymbol{s},1)$

$\hat{\boldsymbol{A}} \in \mathbb{F}_{q^m}[x]^{a\times b}_{<1} \leftarrow \varphi^{-1}(\boldsymbol{A})$ // φ as in (2)
$\hat{\boldsymbol{B}} \leftarrow \mathsf{RightBaseCase}(\hat{\boldsymbol{A}},\boldsymbol{s})$
return $\varphi(\hat{\boldsymbol{B}})$ // mapping φ as in (2)

Theorem 7. *Algorithm 2 is correct and has complexity*

$$O(\rho^{\omega-2}ab)$$

operations over $\mathbb{F}_{q^m}$ where $\rho \leq \min\{a,b\}$ is the rank of $\boldsymbol{A}$.

Proof. Algorithm 2 consists of three parts, which are also illustrated in Figure 1: The first line maps the input matrix $\boldsymbol{A}$ to $\mathbb{F}_{q^m}[x]$; note that this is actually the identity mapping since $\deg \boldsymbol{A} < 1$. Then Lines 1 to 7 apply the well-known PM-basis algorithm [74], [72] over $\mathbb{F}_{q^m}[x]$, and finally, the resulting matrix $\hat{\boldsymbol{B}}$, which is an $\boldsymbol{s}$-ordered weak-Popov approximant basis of $\hat{\boldsymbol{A}}$ of order 1 over $\mathbb{F}_{q^m}[x]$, is mapped back to the skew polynomial ring. We show that $\boldsymbol{B} \in \mathsf{owPopovApprox}_{\mathsf{R}}(\hat{\boldsymbol{A}}, \boldsymbol{s}, 1)$ using properties of $\hat{\boldsymbol{B}}$ and φ.

Note that the mapping φ does not change the degree of a polynomial. As $\hat{\boldsymbol{B}}$ is in $\boldsymbol{s}$-ordered weak Popov form, so is $\boldsymbol{B}$.

Denote by $\boldsymbol{b}_i$ and $\hat{\boldsymbol{b}}_i$ the i-th column of $\boldsymbol{B}$ and $\hat{\boldsymbol{B}}$, respectively. Since $\hat{\boldsymbol{b}}_i$ is a right approximant of $\hat{\boldsymbol{A}}$ and due to Property (4), we have

$$\begin{aligned} \boldsymbol{A}\boldsymbol{b}_i \operatorname{rem_l} x &= \varphi(\hat{\boldsymbol{A}})\varphi(\hat{\boldsymbol{b}}_i) \operatorname{rem_l} x \\ &= \varphi(\hat{\boldsymbol{A}}\hat{\boldsymbol{b}}_i \operatorname{rem} x) = \varphi(\boldsymbol{0}) = \boldsymbol{0}, \end{aligned} \tag{5}$$

so the columns of $\boldsymbol{B}$ are right approximants of $\boldsymbol{A}$ of order 1.

It is left to show that the (right) column space of $\boldsymbol{B}$ contains all right approximants of $\boldsymbol{A}$ of order 1. For this, we identify two key properties of $\boldsymbol{B}$ and $\hat{\boldsymbol{B}}$, respectively.

1) The (right) column space of $x\boldsymbol{I}_b \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$ is contained in the column space of $\boldsymbol{B}$.
2) If $\hat{\boldsymbol{v}} = \hat{\boldsymbol{B}}\hat{\boldsymbol{\lambda}}$ for two vectors $\hat{\boldsymbol{v}}, \hat{\boldsymbol{\lambda}} \in \mathbb{F}_{q^m}[x]^b$ and $\deg \hat{\boldsymbol{v}} = 0$, then $\deg \boldsymbol{\lambda} = 0$.

The first property directly follows from the shape of $\boldsymbol{B}$, which is up to row and column permutations equivalent to a matrix

$$\begin{bmatrix} x\boldsymbol{I}_\rho & \boldsymbol{D} \\ \boldsymbol{0} & \boldsymbol{I}_{b-\rho} \end{bmatrix} \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}, \tag{6}$$

where $\deg \boldsymbol{D} \leq 0$. For the second property, first observe that $\deg \hat{\boldsymbol{\lambda}} = \operatorname{cdeg}_{\boldsymbol{0}} \hat{\boldsymbol{\lambda}} \leq \operatorname{cdeg}_{\operatorname{cdeg}_{\boldsymbol{0}} \boldsymbol{B}} \hat{\boldsymbol{\lambda}}$ since $\operatorname{cdeg}_{\boldsymbol{0}} \boldsymbol{B} \geq 0$. Since (6), seen over $\mathbb{F}_{q^m}[x]$, is in unshifted ($\boldsymbol{s} = \boldsymbol{0}$) ordered weak Popov form, the predictable degree property implies

$$\operatorname{cdeg}_{\operatorname{cdeg}_{\boldsymbol{0}} \boldsymbol{B}} \hat{\boldsymbol{\lambda}} = \operatorname{cdeg}_{\boldsymbol{0}} \hat{\boldsymbol{v}} = \deg \hat{\boldsymbol{v}} = 0.$$

Let now $\boldsymbol{v}$ be a right approximant of $\boldsymbol{A}$ of order 1 and we should show that it is in the column space of $\boldsymbol{B}$. Write $\boldsymbol{v} = \boldsymbol{v}_0 + x\boldsymbol{v}_1$, where $\deg \boldsymbol{v}_0 \leq 0$. By Property 1), then $x\boldsymbol{v}_1$ is in the column space of $\boldsymbol{B}$, so we are done if the same holds for $\boldsymbol{v}_0$. By the same argument as in (5), the vector $\hat{\boldsymbol{v}}_0 := \varphi^{-1}(\boldsymbol{v}_0) \in \mathbb{F}_{q^m}[x]^b$ is a right approximant of $\hat{\boldsymbol{A}}$ and there is a vector $\hat{\boldsymbol{\lambda}} \in \mathbb{F}_{q^m}[x]^b$ such that $\hat{\boldsymbol{v}}_0 = \hat{\boldsymbol{B}}\hat{\boldsymbol{\lambda}}$. Due to Property 2), we have $\deg \hat{\boldsymbol{\lambda}} = 0$, which by (3) implies

$$\boldsymbol{v}_0 = \varphi(\hat{\boldsymbol{v}}_0) = \varphi(\hat{\boldsymbol{B}}\hat{\boldsymbol{\lambda}}) = \varphi(\hat{\boldsymbol{B}})\varphi(\hat{\boldsymbol{\lambda}}) = \boldsymbol{B}\varphi(\hat{\boldsymbol{\lambda}}).$$

Correctness of the algorithm follows.

The main computational task is to compute the row and column rank profile of the matrix $\hat{\boldsymbol{A}} \in \mathbb{F}_{q^m}^{a\times b}$ of rank ρ which requires $O(\rho^{\omega-2}ab)$ operations in $\mathbb{F}_{q^m}$ [75, Thm. 2.10]. □

Remark 8. *We chose to rely on the PM-basis algorithm over $\mathbb{F}_{q^m}[x]$ in the proof of Theorem 7 since it stresses the similarities and differences of the skew and ordinary polynomial case for approximant bases of order 1 of matrices of degree 0. For a self-contained proof of Theorem 7, which directly adapts the key ideas of the PM-basis correctness proof in [72], we refer to the conference version of this paper [1].*

The same reduction is not possible with the mapping φ for higher degrees and orders as we can see in Example 38 (Appendix B).

For the left case, we use the following bijective mapping.

$$\begin{aligned} \psi \;:\; \mathbb{F}_{q^m}[x] &\to \mathbb{F}_{q^m}[x;\sigma], \\ \textstyle\sum_i f_i x^i &\mapsto \textstyle\sum_i f_i x^i. \end{aligned} \tag{7}$$

The resulting algorithm is presented in Algorithm 3 and we prove its correctness in Theorem 9.

Theorem 9. *Algorithm 3 is correct and has complexity*

$$O(\rho^{\omega-2}ab)$$

operations over $\mathbb{F}_{q^m}$, where $\rho \leq \min\{a,b\}$ is the rank of $\boldsymbol{A}$.

Algorithm 3: LeftSkewBaseCase

Input : $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ with $\deg(\boldsymbol{A}) < 1$, $\boldsymbol{s} \in \mathbb{Z}^a$
Output: $\boldsymbol{B} \in \mathsf{owPopovApprox}_{\mathsf{L}}(\boldsymbol{A}, \boldsymbol{s}, 1)$
$\hat{\boldsymbol{A}} \in \mathbb{F}_{q^m}[x]^{a\times b}_{<1} \leftarrow \psi^{-1}(\boldsymbol{A})$ // ψ as in (7)
$\hat{\boldsymbol{B}} \leftarrow \mathsf{RightBaseCase}\big(\hat{\boldsymbol{A}}^\top, \boldsymbol{s}\big)$
return $\psi\big(\hat{\boldsymbol{B}}^\top\big)$ // mapping ψ as in (7)

Proof. The proof is the same as the one of Theorem 7, using the analogous properties of (3) and (4) for ψ in the left side,

$$\psi(fh) = \psi(f)\psi(h) \quad \forall\, f \in \mathbb{F}_{q^m}[x]_{<1} \text{ and } h \in \mathbb{F}_{q^m}[x], \tag{8}$$
$$\psi(fg \bmod_{\mathrm{r}} x) = \psi(f)\psi(g) \bmod_{\mathrm{r}} x \quad \forall\, f, g \in \mathbb{F}_{q^m}[x], \tag{9}$$

as well as the following "transposed" analogs of the properties of $\boldsymbol{B}$ and $\hat{\boldsymbol{B}}$ in the right case:

1) The (left) row space of $x\boldsymbol{I}_a \in \mathbb{F}_{q^m}[x;\sigma]^{a\times a}$ is contained in the row space of $\boldsymbol{B}$.
2) If $\hat{\boldsymbol{v}} = \hat{\boldsymbol{\lambda}}\hat{\boldsymbol{B}}$ for two vectors $\hat{\boldsymbol{v}}, \hat{\boldsymbol{\lambda}} \in \mathbb{F}_{q^m}[x]^a$ and $\deg \hat{\boldsymbol{v}} = 0$, then $\deg \boldsymbol{\lambda} = 0$.

Recall that over $\mathbb{F}_{q^m}[x]$, the transpose of a right approximant basis of $\hat{\boldsymbol{A}}^\top$ is a left approximant basis of $\hat{\boldsymbol{A}}$. □

2) Recursive Algorithm: Right and Left PM-Basis: This section presents a skew-polynomial variant of the PM-basis algorithm, which computes approximant bases of higher order $d > 1$ in a recursive fashion using Algorithm 2 (right side) and Algorithm 3 (left side) as its base case, respectively. The recursion step is based on the following lemmas, which would remain true if stated over ordinary polynomial rings. However the ordering of the involved polynomial products and the choice of left/right modulo is central for the statements and proofs to hold over the non-commutative skew polynomial ring.

Lemma 10. *Let $d \in \mathbb{Z}_{>0}$, $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ of degree less than d, and $d_1, d_2 \in \mathbb{Z}_{>0}$ be such that $d_1 + d_2 = d$.*

Let $\boldsymbol{s} \in \mathbb{Z}^b$, $\boldsymbol{B}_1 \in \mathsf{owPopovApprox}_{\mathsf{R}}(\boldsymbol{A} \bmod_{\mathrm{l}} x^{d_1}, \boldsymbol{s}, d_1)$, and $\boldsymbol{B}_2 \in \mathsf{owPopovApprox}_{\mathsf{R}}(x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \bmod_{\mathrm{l}} x^{d-d_1}, \boldsymbol{t}, d_2)$, where $\boldsymbol{t} := \mathrm{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$. Then, $\boldsymbol{B}_1\boldsymbol{B}_2 \in \mathsf{owPopovApprox}_{\mathsf{R}}(\boldsymbol{A}, \boldsymbol{s}, d)$.

Let $\boldsymbol{s} \in \mathbb{Z}^a$, $\boldsymbol{B}_1 \in \mathsf{owPopovApprox}_{\mathsf{L}}(\boldsymbol{A} \bmod_{\mathrm{l}} x^{d_1}, \boldsymbol{s}, d_1)$, and $\boldsymbol{B}_2 \in \mathsf{owPopovApprox}_{\mathsf{L}}(\boldsymbol{B}_1\boldsymbol{A}x^{-d_1} \bmod_{\mathrm{r}} x^{d-d_1}, \boldsymbol{t}, d_2)$, where $\boldsymbol{t} := \mathrm{rdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$. Then, $\boldsymbol{B}_2\boldsymbol{B}_1 \in \mathsf{owPopovApprox}_{\mathsf{L}}(\boldsymbol{A}, \boldsymbol{s}, d)$.

Proof: We prove the right case, the left side follows analogously. First, we show that all approximants of $\boldsymbol{A}$ of order d are right $\mathbb{F}_{q^m}[x;\sigma]$-linear combinations of the columns of $\boldsymbol{B}_1\boldsymbol{B}_2$. Let $\boldsymbol{b}$ be an approximant of $\boldsymbol{A}$ of order d and decompose $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{A} \bmod_{\mathrm{l}} x^{d_1} + x^d\tilde{\boldsymbol{A}}$. Then,

$$\begin{aligned}&(\boldsymbol{A} \bmod_{\mathrm{l}} x^{d_1} + x^d\tilde{\boldsymbol{A}})\boldsymbol{b} \equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^d\\ \implies\quad &(\boldsymbol{A} \bmod_{\mathrm{l}} x^{d_1})\boldsymbol{b} \equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^{d_1}.\end{aligned}$$

Hence, $\boldsymbol{b}$ is also an approximant of $\boldsymbol{A} \bmod_{\mathrm{l}} x^{d_1}$ of order d_1 and we can write $\boldsymbol{b} = \boldsymbol{B}_1\boldsymbol{\lambda}$ for some $\boldsymbol{\lambda} \in \mathbb{F}_{q^m}[x;\sigma]^b$. This $\boldsymbol{\lambda}$ again fulfills

$$\begin{aligned}&\boldsymbol{A}\boldsymbol{B}_1\boldsymbol{\lambda} \equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^d,\\ \implies &\boldsymbol{A}\boldsymbol{B}_1\boldsymbol{\lambda} = x^d\boldsymbol{v}'\\ \implies &x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1\boldsymbol{\lambda} = x^{d-d_1}\boldsymbol{v}' = x^{d_2}\boldsymbol{v}'\\ \implies &x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1\boldsymbol{\lambda} \equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^{d_2},\end{aligned}$$

for some $\boldsymbol{v}' \in \mathbb{F}_{q^m}[x;\sigma]$. Again, we can decompose

$$x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 = (x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \bmod_{\mathrm{l}} x^{d_2}) + x^{d_2}\tilde{\boldsymbol{A}}$$

and have $(x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \bmod_{\mathrm{l}} x^{d_2})\boldsymbol{\lambda} \equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^{d_2}$. Thus, $\boldsymbol{\lambda}$ is an approximant of $x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \bmod_{\mathrm{l}} x^{d_2}$ of order d_2 and can be written as $\boldsymbol{\lambda} = \boldsymbol{B}_2\boldsymbol{\mu}$. Overall, we get

$$\boldsymbol{\lambda} = \boldsymbol{B}_1\boldsymbol{B}_2\boldsymbol{\mu},$$

so $\boldsymbol{\lambda}$ is in the right column span of $\boldsymbol{B}_1\boldsymbol{B}_2$.

For the other direction, let $\boldsymbol{b} = \boldsymbol{B}_1\boldsymbol{B}_2\boldsymbol{\mu}$ be in the column span of $\boldsymbol{B}_1\boldsymbol{B}_2$. We show that $\boldsymbol{b}$ is an approximant of $\boldsymbol{A}$ of order d. Let $\boldsymbol{\lambda} = \boldsymbol{B}_2\boldsymbol{\mu}$. Thus, $\boldsymbol{\lambda}$ is an approximant of $\boldsymbol{x}^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \bmod_{\mathrm{l}} x^{d_2}$ and we have

$$\begin{aligned}&(\boldsymbol{x}^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \bmod_{\mathrm{l}} x^{d_2})\boldsymbol{\lambda} \equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^{d_2}\\ \implies &(\boldsymbol{x}^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \bmod_{\mathrm{l}} x^{d_2})\boldsymbol{\lambda} = x^{d_2}\boldsymbol{v}'\end{aligned}$$

for some $\boldsymbol{v}' \in \mathbb{F}_{q^m}[x;\sigma]^b$. We can again write $\boldsymbol{x}^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 \operatorname{rem}_{\mathrm{l}} x^{d_2} = \boldsymbol{x}^{-d_1}\boldsymbol{A}\boldsymbol{B}_1 - x^{d_2}\tilde{\boldsymbol{A}}$ and get

$$\begin{aligned}\boldsymbol{x}^{-d_1}\boldsymbol{A}\boldsymbol{B}_1\boldsymbol{\lambda} &= x^{d_2}\boldsymbol{v}' + x^{d_2}\tilde{\boldsymbol{A}}\boldsymbol{\lambda}\\ \implies \boldsymbol{A}\boldsymbol{B}_1\boldsymbol{\lambda} &= x^{d_1+d_2}\boldsymbol{v}' + x^{d_1+d_2}\tilde{\boldsymbol{A}}\boldsymbol{\lambda}\\ \implies \boldsymbol{A}\boldsymbol{B}_1\boldsymbol{\lambda} &\equiv \boldsymbol{0} \bmod_{\mathrm{l}} x^d.\end{aligned}$$

Hence, $\boldsymbol{b} = \boldsymbol{B}_1\boldsymbol{B}_1\boldsymbol{\mu}$ is an approximant of $\boldsymbol{A}$ of order d.

By Lemma 3, $\boldsymbol{B}_1\boldsymbol{B}_2$ is in $\boldsymbol{s}$-ordered weak Popov form and the statement follows. ∎

Algorithms 4 and 5 are fast divide & conquer algorithms for constructing right and left approximant bases over skew polynomial rings, respectively. The algorithms use Lemma 10 with $d_1 = \lceil d/2 \rceil$ and $d_2 = d - d_1$ recursively and are fast skew variants of [74, PM-Basis].

Algorithm 4: RightSkewPMBasis

Input :
- positive integer $d \in \mathbb{Z}_{>0}$,
- matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ of degree $< d$,
- shifts $\boldsymbol{s} \in \mathbb{Z}^b$.

Output: $\boldsymbol{B} \in \mathsf{owPopovApprox}_{\mathsf{R}}(\boldsymbol{A}, \boldsymbol{s}, d)$

1 **if** $d = 1$ **then**
2 **return** $\mathsf{RightSkewBaseCase}(\boldsymbol{A}, \boldsymbol{s})$ // Algorithm 2
3 **else**
4 $d_1 \leftarrow \lceil d/2 \rceil$, $d_2 \leftarrow d - d_1$
5 $\boldsymbol{B}_1 \leftarrow \mathsf{RightSkewPMBasis}\left(d_1, \boldsymbol{A} \operatorname{rem}_{\mathrm{l}} x^{d_1}, \boldsymbol{s}\right)$
6 $\boldsymbol{G} \leftarrow \left(x^{-d_1}\boldsymbol{A}\boldsymbol{B}_1\right) \operatorname{rem}_{\mathrm{l}} x^{d_2}$; $\boldsymbol{t} \leftarrow \operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$
7 $\boldsymbol{B}_2 \leftarrow \mathsf{RightSkewPMBasis}\left(d_2, \boldsymbol{G}, \boldsymbol{t}\right)$
8 **return** $\boldsymbol{B}_1\boldsymbol{B}_2$

Algorithm 5: LeftSkewPMBasis

Input :
- positive integer $d \in \mathbb{Z}_{>0}$,
- matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ of degree $< d$,
- shifts $\boldsymbol{s} \in \mathbb{Z}^a$.

Output: $\boldsymbol{B} \in \mathsf{owPopovApprox}_{\mathsf{L}}(\boldsymbol{A}, \boldsymbol{s}, d)$

1 **if** $d = 1$ **then**
2 **return** $\mathsf{LeftSkewBaseCase}(\boldsymbol{A}, \boldsymbol{s})$ // Algorithm 3
3 **else**
4 $d_1 \leftarrow \lceil d/2 \rceil$, $d_2 \leftarrow d - d_1$
5 $\boldsymbol{B}_1 \leftarrow \mathsf{LeftSkewPMBasis}\left(d_1, \boldsymbol{A} \operatorname{rem}_{\mathrm{r}} x^{d_1}, \boldsymbol{s}\right)$
6 $\boldsymbol{G} \leftarrow \left(\boldsymbol{B}_1\boldsymbol{A}x^{-d_1}\right) \operatorname{rem}_{\mathrm{r}} x^{d_2}$; $\boldsymbol{t} \leftarrow \operatorname{rdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$
7 $\boldsymbol{B}_2 \leftarrow \mathsf{LeftSkewPMBasis}\left(d_2, \boldsymbol{G}, \boldsymbol{t}\right)$
8 **return** $\boldsymbol{B}_2\boldsymbol{B}_1$

Theorem 11. *Algorithm 4 is correct and has complexity*

$$\tilde{O}\big(\max\{a,b\}b^{\omega-1}\mathcal{M}_{q,m}(d)\big).$$

Algorithm 5 is correct and has complexity

$$\tilde{O}\big(a^{\omega-1}\max\{a,b\}\mathcal{M}_{q,m}(d)\big).$$

Proof. Correctness follows from Lemma 10, as well as the correctness of the base cases (Theorem 7 for Algorithm 2 and Theorem 9 for Algorithm 3).

As for the complexity, the algorithms calls themselves twice with input size $\approx d/2$. Taking a matrix left or right modulo x^{d_i} corresponds to setting all coefficients of degree at least d_i to zero in each entry. Multiplying x^{-d_1} from the left in Line 6 of Algorithm 4 requires to apply an automorphism to each polynomial coefficient, hence costs $O(\ell^2 d)$ operations over $\mathbb{F}_{q^m}$. Note that this is not necessary in Algorithm 5 since the monomial is multiplied from the right.

The any other costful operations are the base cases and the matrix multiplications (Lines 6 and 8 in Algorithm 4 and Lines 6 and 8 in Algorithm 4). We discuss the right case, the other side follows analogously by replacing a and b in the complexity expression. The two multiplications are $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ times $\boldsymbol{B}_1 \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$ and $\boldsymbol{B}_1 \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$ times $\boldsymbol{B}_2 \in \mathbb{F}_{q^m}[x;\sigma]^{b\times b}$, all matrices have degree at most d. The product $\boldsymbol{A}\boldsymbol{B}_1$ can be computed in $O(\frac{a}{b}b^\omega \mathcal{M}_{q,m}(d)) = O(ab^{\omega-1}\mathcal{M}_{q,m}(d))$ if $a \geq b$ and in $O(b^\omega \mathcal{M}_{q,m}(d))$ otherwise. The product $\boldsymbol{B}_1\boldsymbol{B}_2$ costs $O(b^\omega \mathcal{M}_{q,m}(d))$. In total, the matrix multiplications can be computed with complexity $O(\max\{a,b\}b^{\omega-1}\mathcal{M}_{q,m}(d))$. The base case, Algorithm 2, costs $O(\min\{a,b\}^{\omega-1}ab)$.

Hence, we obtain the claimed complexity by the master theorem for divide-and-conquer recurrences. □

Remark 12. *Using Lemma 10 with $d_1 = 1$ (i.e. the base case) and $d_2 = d-1$ in an iterative manner results in right and left skew variants of [74, M-Basis] where the order of $\boldsymbol{B}$ is increased by one in each iteration. The complexities of the resulting algorithms are $\tilde{O}\big(\max\{a,b\}b^{\omega-1}d^2\big)$ (right case) and $\tilde{O}\big(a^{\omega-1}\max\{a,b\}d^2\big)$ (left case) operations in $\mathbb{F}_{q^m}$, respectively.*

This is asymptotically slower than Algorithms 4 and 5 using skew polynomial multiplication algorithms of sub-quadratic complexity over $\mathbb{F}_{q^m}$, e.g. [69], [70]. In particular, applying the skew M-basis algorithm to the decoding problems in the remainder of the paper would not improve the asymptotic costs of the state-of-the-art decoder implementations.

However, for small orders d, the skew M-basis algorithm might be faster than the skew PM-basis algorithm due to large hidden constants in the asymptotic expressions of asymptotically fast skew polynomial multiplication algorithms. The two methods can also be combined by calling M-basis (instead of PM-basis) inside PM-basis as soon as d is small enough.

For completeness, we present the skew M-basis algorithm and prove its complexity in Appendix A.

IV. Fast Decoding of Rank-Metric and Subspace Codes

We show how to speed up interpolation-based decoding of interleaved Gabidulin codes in the rank metric (Wachter-Zeh–Zeh decoder [5]) and lifted interleaved Gabidulin codes in the subspace metric (Bartz–Wachter-Zeh decoder [6]).

The interpolation and root-finding steps of both considered decoders are special instances of two general computational problems, which we state and relate to the decoders. Then we present new algorithms to solve the two problems by reducing Problem 13 to computing a *left* approximant basis (Algorithm 6 in Section IV-B), and show that Problem 14 (i.e., root finding) can be efficiently solved by a *right* approximant basis (Algorithm 7 in Section IV-C).

In this section, we only use the operator evaluation of skew polynomials (cf. Section II-C).

A. Computational Problems and their Relation to Decoding

Problem 13 (Vector (Operator) Interpolation)**.** *Given $\ell, n, D \in \mathbb{Z}_{>0}$, $\boldsymbol{w} \in \mathbb{Z}_{\geq 0}^{\ell+1}$, and $\boldsymbol{U} = [U_{i,j}] \in \mathbb{F}_{q^m}^{n\times(\ell+1)}$ whose rows (called "interpolation points") are $\mathbb{F}_q$-linearly independent. Consider the $\mathbb{F}_{q^m}$-vector space $\mathcal{Q}$ (left scalar multiplication) of vectors $\boldsymbol{Q} = [Q_0, Q_1, \ldots, Q_\ell] \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1}$ that satisfy the following two conditions:*

$$\sum_{j=1}^{\ell+1} Q_{j-1}(U_{i,j}) = 0, \qquad \forall\, i = 1, \ldots, n, \tag{10}$$

$$\operatorname{rdeg}_{\boldsymbol{w}}(\boldsymbol{Q}) < D. \tag{11}$$

Find left $\mathbb{F}_{q^m}[x;\sigma]$-linearly independent $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell')} \in \mathcal{Q} \setminus \{\boldsymbol{0}\}$ whose left $\mathbb{F}_{q^m}[x;\sigma]$-span contains $\mathcal{Q}$.

Problem 14 (Vector Root Finding)**.** *Given $\ell, n \in \mathbb{Z}_{>0}$, $\boldsymbol{k} \in \mathbb{Z}_{>0}^{\ell}$, and vectors $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell')} \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1} \setminus \{\boldsymbol{0}\}$ that are left $\mathbb{F}_{q^m}[x;\sigma]$-linearly independent (this implies $\ell' \leq \ell+1$) and fulfill $\deg \boldsymbol{Q}^{(i)} \leq n$ for all i. Find a basis of the affine $\mathbb{F}_{q^m}$-vector space (scalar multiplication from the right)*

$$\mathcal{R} := \big\{[f^{(1)}, \ldots, f^{(\ell)}] \in \mathbb{F}_{q^m}[x;\sigma]^{\ell} \;:\; Q_0^{(i)} + \textstyle\sum_{j=1}^{\ell} Q_j^{(i)} f^{(j)} = 0 \,\forall\, i,\ \deg f^{(j)} < k^{(j)} \,\forall\, j\big\}. \tag{12}$$

Complexity-wise, we consider only the cases $D \in \Theta(n)$ and $\max_i k^{(i)} \in \Theta(n)$ since they are the most relevant for decoding. See Section VI-C for a discussion on the cases $D, \max_i k^{(i)} \ll n$ and $D, \max_i k^{(i)} \gg n$. The fastest algorithm to solve Problem 13 with $D \in \Theta(n)$ is [8] with a complexity of $O(\ell^2 n^2)$ over $\mathbb{F}_{q^m}$. If the first column of $\boldsymbol{U}$ consists of $\mathbb{F}_q$-linearly independent elements (see, e.g., Wachter-Zeh decoder [5] below), Problem 13 can be solved with complexity $O(\ell^3 \mathcal{M}_{q,m}(\ell n))$ [9], [71]. For $\max_i k^{(i)} \in \Theta(n)$, Problem 14 can be solved in $O(\ell^3 n^2)$ over $\mathbb{F}_{q^m}$ [5] or, if $|\mathcal{R}| = 1$, in $O(\ell^2 n^2)$ [6].

1) Interpolation-Based Decoding of Rank-Metric Codes: We recall the Wachter-Zeh–Zeh decoder and connect it to Problems 13 and 14. Let $n \leq m$ and ℓ be positive integers, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n] \in \mathbb{F}_{q^m}^n$ be a vector whose entries are linearly independent over $\mathbb{F}_q$, and $\boldsymbol{k} = [k^{(1)}, \dots, k^{(\ell)}] \in \{1, \dots, n\}^\ell$. The corresponding interleaved Gabidulin code [19] is

$$\mathcal{IC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n, \boldsymbol{k}] := \left\{ \begin{bmatrix} f^{(1)}(\alpha_1) & \cdots & f^{(1)}(\alpha_n) \\ \vdots & \ddots & \vdots \\ f^{(\ell)}(\alpha_1) & \cdots & f^{(\ell)}(\alpha_n) \end{bmatrix} : f^{(i)} \in \mathbb{F}_{q^m}[x;\sigma]_{<k^{(i)}} \, \forall \, i \right\}.$$

All codewords $\boldsymbol{C}$, which are $\mathbb{F}_{q^m}^{\ell \times n}$ matrices, have a corresponding message polynomial vector $\boldsymbol{f} := \left[f^{(1)}, \dots, f^{(\ell)}\right]$, whose entries evaluate to the rows of $\boldsymbol{C}$ at $\boldsymbol{\alpha}$.

The codes are designed for the following generalization of the rank metric. Fix a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then any element of $\mathbb{F}_{q^m}$ can be written as a vector in $\mathbb{F}_q^m$ by expanding the element in this basis. The rank weight $\mathrm{wt_R}(\boldsymbol{A})$ of a matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}^{\ell \times n}$ is the $\mathbb{F}_q$ rank of the matrix in $\mathbb{F}_q^{\ell m \times n}$ that we obtain by expanding each entry of $\boldsymbol{A}$ into a column vector. The rank distance of two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{F}_{q^m}^{\ell \times n}$ is the rank weight of their difference, i.e., $\mathrm{d_R}(\boldsymbol{A}, \boldsymbol{B}) := \mathrm{wt_R}(\boldsymbol{A} - \boldsymbol{B})$. Due to $\mathbb{F}_q^{\ell m \times n} \simeq \mathbb{F}_{q^{m\ell}}^n$, this is the usual rank metric in $\mathbb{F}_{q^{m\ell}}^n$. See Section I-B for applications of the codes and the metric.

Let $\mathcal{IC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n, \boldsymbol{k}]$ be an interleaved Gabidulin code and $\boldsymbol{R} \in \mathbb{F}_{q^m}^{\ell \times n}$ be a received word. The interpolation step of the Wachter-Zeh–Zeh decoder solves Problem 13 with input ℓ, n,

$$\begin{aligned} D &= n - \left\lceil \tfrac{\ell(n+1) - \sum_{i=1}^{\ell} k^{(i)}}{\ell+1} \right\rceil + 1, \\ \boldsymbol{w} &= [0, k^{(1)} - 1, \dots, k^{(\ell)} - 1] \in \mathbb{Z}_{\geq 0}^{\ell+1}, \quad \text{and} \\ \boldsymbol{U} &= \begin{bmatrix} \boldsymbol{\alpha}^\top & \boldsymbol{R}^\top \end{bmatrix} \in \mathbb{F}_{q^m}^{n \times (\ell+1)}. \end{aligned} \tag{13}$$

This instance of the problem always has a non-trivial solution (i.e., $\mathcal{Q} \neq \{0\}$). If the output[1] of this problem is input to Problem 14 (root-finding step), then the space $\mathcal{R}$ in Problem 14 contains all message polynomial vectors $\boldsymbol{f} \in \mathbb{F}_{q^m}[x;\sigma]^\ell$ of codewords $\boldsymbol{C} \in \mathcal{IC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n, \boldsymbol{k}]$ whose rank distance to the received words is smaller than

$$\mathrm{d_R}(\boldsymbol{C}, \boldsymbol{R}) < \tfrac{\ell}{\ell+1} \left(n - \tfrac{1}{\ell} \textstyle\sum_i k^{(i)} + 1\right).$$

This gives a list decoder with list size at most $|\mathcal{R}|$ ($\mathcal{R}$ may contain vectors that do not correspond to codewords lying within the decoding radius). Wachter-Zeh and Zeh derived an exponential upper bound on $|\mathcal{R}|$ and a bound (which is close to 1 for many parameters) on the expected size of $|\mathcal{R}|$ for a received word $\boldsymbol{R}$ that is chosen uniformly at random from $\mathbb{F}_{q^m}^{\ell \times n}$.[2] The algorithm can be turned into a partial unique decoder by declaring a failure for $|\mathcal{R}| > 1$.

The previous-fastest realization of the decoder has complexity $O(\ell^2 n^2)$. With the new algorithms to solve Problems 13 and 14 in the next subsections, we get the following speed-up.

Theorem 15. *Decoding an interleaved Gabidulin code $\mathcal{IC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n, \boldsymbol{k}]$ using the decoder in [5], where*

- *the interpolation step is implemented using Algorithm 6 (Section IV-B) with input ℓ, n, D, $\boldsymbol{w}$, $\boldsymbol{U}$ as in (13) and*
- *the root-finding step is implemented using Algorithm 7 (Section IV-C) with input ℓ, n, $\boldsymbol{k}$, and the output $\boldsymbol{Q}^{(1)}, \dots, \boldsymbol{Q}^{(\ell')}$ of the interpolation step,*

has complexity $\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$ operations over $\mathbb{F}_q$.

Proof. Correctness and complexity follow directly from the correctness and complexity of Algorithm 6 (Theorem 22) and Algorithm 7 (Theorem 25) and the results in [5] (see also the brief summary above). We only need to be careful about two points: the entries of the first column of $\boldsymbol{U}$ are $\mathbb{F}_q$-linearly independent by definition of the α_i; also, $\boldsymbol{Q}^{(1)}, \dots, \boldsymbol{Q}^{(\ell')}$ is a valid input to Algorithm 7 since by the choice of D and the degree constraint in Problem 13, we have $\deg \boldsymbol{Q}^{(i)} \leq n$. □

2) Interpolation-Based Decoding of Subspace Codes: We recall the Bartz–Wachter-Zeh decoder [6]. Let $n_t \leq m$ and ℓ be positive integers, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{n_t}] \in \mathbb{F}_{q^m}^{n_t}$ be a vector whose entries are linearly independent over $\mathbb{F}_q$, and $\boldsymbol{k} = [k^{(1)}, \dots, k^{(\ell)}] \in \{1, \dots, n_t\}^\ell$. The corresponding lifted interleaved Gabidulin code [15] is defined as

$$\mathcal{LIC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n_t, \boldsymbol{k}] := \left\{ \left\langle \begin{bmatrix} \boldsymbol{\alpha}^\top & \boldsymbol{C}^\top \end{bmatrix} \right\rangle_q : \boldsymbol{C} \in \mathcal{IC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n_t, \boldsymbol{k}] \right\}$$

[1] Our interpolation problem output differs slightly from [5], where either one solution or an $\mathbb{F}_{q^m}$-basis of $\mathcal{Q}$ is found. It is easy to see that a set of $\mathbb{F}_{q^m}[x;\sigma]$-linearly independent vectors whose span contains $\mathcal{Q}$ does not change the root space $\mathcal{R}$ compared to a full $\mathbb{F}_{q^m}$-basis. Further, Problem 13 and Algorithm 6 in Section IV-B can be easily adapted to output one solution.

[2] The proof of [5, Lemma 6] derives a bound on the expected size of $|\mathcal{R}|$ for a uniformly chosen received word $\boldsymbol{R}$. However, the lemma statement does not fit to the proof since it assumes that $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$ for a codeword $\boldsymbol{C}$ and error $\boldsymbol{E}$ of weight at most a given value τ, i.e., depending on the code and τ, $\boldsymbol{R}$ cannot even attain all values of $\mathbb{F}_{q^m}^{\ell \times n}$.

where $\left\langle \begin{bmatrix} \boldsymbol{\alpha}^\top & \boldsymbol{C}^\top \end{bmatrix} \right\rangle_q$ denotes the $\mathbb{F}_q$-linear row space of the matrix from $\mathbb{F}_q^{n_t \times m(\ell+1)}$ obtained by expanding each entry of the matrix $\begin{bmatrix} \boldsymbol{\alpha}^\top & \boldsymbol{C}^\top \end{bmatrix} \in \mathbb{F}_{q^m}^{n \times (\ell+1)}$ into a $1 \times m$ row vector over $\mathbb{F}_q$ using a fixed basis of $\mathbb{F}_{q^m}$. Hence, codewords are n_t-dimensional subspaces of $\mathbb{F}_q^{m(\ell+1)}$. The *subspace distance* between two subspaces $\mathcal{U}, \mathcal{V}$ of $\mathbb{F}_q^{m(\ell+1)}$ is defined as

$$d_s(\mathcal{U}, \mathcal{V}) = \dim(\mathcal{U}) + \dim(\mathcal{V}) - 2\dim(\mathcal{U} \cap \mathcal{V}). \tag{14}$$

This is a natural metric in the *operator channel* [22], which for an input subspace $\mathcal{V}$ of $\dim(\mathcal{V}) = n_t$ returns a subspace

$$\mathcal{U} = \mathcal{H}_{n_t - \delta}(\mathcal{V}) \oplus \mathcal{E}, \tag{15}$$

where $\mathcal{H}_{n_t-\delta}(\mathcal{V})$ is a $(n_t - \delta)$-dimensional subspace of $\mathcal{V}$, and $\mathcal{E}$ denotes an error space of dimension γ with $\mathcal{V} \cap \mathcal{E} = \{\boldsymbol{0}\}$. We call γ the number of *insertions* and δ the number of *deletions*. Hence, the received space $\mathcal{U}$ has dimension

$$n_r := \dim(\mathcal{U}) = n_t - \delta + \gamma. \tag{16}$$

We say that a subspace $\mathcal{V}$ is (γ, δ)-*reachable* from a subspace $\mathcal{U}$ if there exists a realization of the operator channel (15) with γ insertions and δ deletions that transforms the input $\mathcal{V}$ to the output $\mathcal{U}$. If a space $\mathcal{V}$ is (γ, δ)-reachable from a space $\mathcal{U}$, then we have that $d_s(\mathcal{U}, \mathcal{V}) = \gamma + \delta$. See Section I-B for applications of the codes and the metric.

Let $\mathcal{LIC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n_t, \boldsymbol{k}]$ be a lifted interleaved Gabidulin code and $\mathcal{U} \subseteq \mathbb{F}_q^{m(\ell+1)}$ of dimension $\dim(\mathcal{U}) = n_r$ be a received subspace, given in form of a basis $\boldsymbol{U} \in \mathbb{F}_{q^m}^{n_r \times (\ell+1)}$ with $\mathcal{U} = \langle \boldsymbol{U} \rangle_q$. The interpolation step of the Bartz–Wachter-Zeh decoder asks for a solution $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell)}$ to Problem 13 with input ℓ, $n = n_r$, the basis $\boldsymbol{U} \in \mathbb{F}_{q^m}^{n_r \times (\ell+1)}$,

$$D = \left\lceil \frac{n_r + \sum_{i=1}^{\ell} k^{(i)} - \ell + 1}{\ell + 1} \right\rceil, \qquad \text{and}$$
$$\boldsymbol{w} = [0, k^{(1)} - 1, \ldots, k^{(\ell)} - 1] \in \mathbb{Z}_{\geq 0}^{\ell+1}. \tag{17}$$

Due to the choice of D, this problem instance always has a solution (i.e., $\mathcal{Q} \neq \{0\}$), cf. [6]. The root-finding step consists of solving Problem 14 with input $\ell, n = n_t, \boldsymbol{k}$, as well the $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell)}$ computed above. Then, the space $\mathcal{R}$ contains all message polynomial vectors $\boldsymbol{f} \in \mathbb{F}_{q^m}[x; \sigma]^\ell$ corresponding to codewords $\mathcal{V} \in \mathcal{LIC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n_t, \boldsymbol{k}]$ that are (γ, δ)-reachable from the received space $\mathcal{U}$ with $\dim(\mathcal{U}) = n_r = n_t - \delta + \gamma$ for all γ and δ satisfying[3]

$$\gamma + \ell\delta < \ell\left(n_t - \overline{k} + 1\right). \tag{18}$$

This gives a list decoder with list size at most $|\mathcal{R}|$.

Similar to interleaved Gabidulin codes there exists an upper bound on $|\mathcal{R}|$ (which is exponential in the code parameters) and a bound on the expected size[4] of $|\mathcal{R}|$ (which is close to 1 for many parameters) for a received word $\boldsymbol{R}$ that is drawn uniformly at random from the set of n_r-dimensional subspaces of $\mathbb{F}_q^{m(\ell+1)}$, see [6], [76]. The algorithm can also be interpreted as a probabilistic unique decoder by declaring a decoding failure if $|\mathcal{R}| > 1$, cf. [6].

Using the new algorithms to solve Problems 13 and 14 in the next subsections, we can reduce the complexity of the decoder from $O(\ell^2 \max\{n_r, n_t\}^2)$ [6] to the following expression.

Theorem 16. *Decoding a received subspace of dimension n_r in a lifted interleaved Gabidulin code $\mathcal{LIC}_{\mathrm{Gab}}[\ell, \boldsymbol{\alpha}; n_t, \boldsymbol{k}]$ using the decoder in [6], where*

- *the interpolation step is implemented using Algorithm 6 (Section IV-B) with input ℓ, n_r, D, $\boldsymbol{w}$ as in (17), and a basis $\boldsymbol{U}$ of the received space and*
- *the root-finding step is implemented using Algorithm 7 (Section IV-C) with input ℓ, n_r, $\boldsymbol{k}$, and the output $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell')}$ of the interpolation step,*

has complexity $\tilde{O}\big(\ell^\omega \mathcal{M}_{q,m}(\max\{n_t, n_r\}) + \ell m n_r^{\omega-1}\big)$ operations over $\mathbb{F}_q$.

Proof. Correctness follows from the correctness of Algorithm 6 (Theorem 22) and Algorithm 7 (Theorem 25) and the results in [6] (see also the brief summary above). Note that the vectors $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell')}$ are a valid input to Algorithm 7 since $\deg \boldsymbol{Q}^{(i)} \leq n_r$ by Problem 13.

The complexity is $\tilde{O}\big(\ell^\omega \mathcal{M}_{q,m}(D + n) + \ell m n^{\omega-1}\big)$ for the interpolation step and $\tilde{O}\big(\ell^\omega \mathcal{M}_{q,m}(n + \max_i k^{(i)})\big)$ for the root-finding step by Theorems 22 and 25, respectively. The input variables $n, D, \boldsymbol{k}$ of the two computational problems are connected to the code and channel parameters n_t, n_r as follows. We have $n = n_r$, $D \in O\big(\max\big\{n, \max_i k^{(i)}\big\}\big)$, and $\max_i\{k^{(i)}\} \leq n_t$, which implies the dependency on $\max\{n_t, n_r\}$. □

[3] Due to $d_s(\mathcal{U}, \mathcal{V}) \leq \gamma + \ell\delta$, all $\boldsymbol{f}$ of codewords with $d_s(\mathcal{U}, \mathcal{V}) < \ell(n_t - \overline{k} + 1)$ are in $\mathcal{R}$, but this is a weaker condition than (18).

[4] As in the Wachter-Zeh–Zeh decoder, drawing a received word uniformly at random usually does not correspond to choosing a codeword and a low-weight error uniformly at random, and hence this result is not directly applicable to most channels considered in the literature.

B. A New Algorithm for the Interpolation Step

We relate the interpolation step (Problem 13) to finding a left approximant bases of a matrix $\boldsymbol{A}$ that is constructed from (operator) interpolation and annihilator polynomials depending on the interpolation points (i.e., the input matrix $\boldsymbol{U}$ of the problem).

To construct the matrix $\boldsymbol{A}$, we first need to transform the interpolation points as in the following lemma. Note that we apply $\mathbb{F}_q$-linear elementary row operations to $\boldsymbol{U}$, which due to the $\mathbb{F}_q$-linearity of skew polynomials does not change the interpolation condition, (10), of Problem 13.

Lemma 17. *Consider an instance of Problem 13. Using $\mathbb{F}_q$-linear elementary row operations, we can transform $\boldsymbol{U}$ into a matrix of the form*

$$\boldsymbol{U}' = \left[\begin{array}{c|c} \mathbf{0}_{\nu_1 \times a_1} & \boldsymbol{U}^{(1)} \\ \hline \mathbf{0}_{\nu_2 \times a_2} & \boldsymbol{U}^{(2)} \\ \hline \mathbf{0}_{\nu_3 \times a_3} & \boldsymbol{U}^{(3)} \\ \hline \multicolumn{2}{c}{\vdots} \\ \hline \mathbf{0}_{\nu_\varrho \times a_\varrho} & \boldsymbol{U}^{(\varrho)} \end{array}\right], \tag{19}$$

where $1 \le \varrho \le \ell+1$ and we have $\boldsymbol{U}^{(i)} \in \mathbb{F}_{q^m}^{\nu_i \times (\ell+1-a_i)}$ for $i = 1, \dots, \varrho$, with

- $0 \le a_1 < a_2 < \cdots < a_\varrho < \ell+1$,
- $1 \le \nu_i \le n$ *such that* $\sum_{i=1}^{\varrho} \nu_i = n$, *and*
- *the entries of the first column of $\boldsymbol{U}^{(i)}$ are linearly independent over $\mathbb{F}_q$ for each i.*

The matrix $\boldsymbol{U}'$ can be obtained with $O\big(\ell m n^{\omega-1}\big)$ operations over $\mathbb{F}_q$.

Proof. This can be done by expanding each entry of $\boldsymbol{U} \in \mathbb{F}_{q^m}^{n \times (\ell+1)}$ into a row vector over $\mathbb{F}_q$ of length m, by transforming this $n \times m(\ell+1)$ matrix into row echelon form, and then mapping the resulting matrix back to an $n \times (\ell+1)$ matrix over $\mathbb{F}_{q^m}$. The structure of $\boldsymbol{U}'$ then follows immediately from the row echelon form of the expanded matrix (e.g., the width ν_i of the matrix $\boldsymbol{U}^{(i)}$ will be the number of pivots in the columns $a_i m + 1, \dots, (a_i+1)m$ of the expanded matrix). There will be no zero rows since the rows of $\boldsymbol{U}$ are $\mathbb{F}_q$-linearly independent. The complexity follows by [75, Theorem 2.10]. □

The following lemmas connect Problem 13 to a problem of computing an approximant basis. Since the first columns of all the matrices $\boldsymbol{U}^{(i)}$ are $\mathbb{F}_q$-linearly independent, the polynomials $G^{(i)}$ and $R_j^{(i)}$ in the following lemma are well-defined.

Lemma 18. *Let $\boldsymbol{U}^{(1)}, \dots, \boldsymbol{U}^{(\varrho)}$ be defined as in Lemma 17. Then, $\boldsymbol{Q} = [Q_0, \dots, Q_\ell] \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1}$ satisfies Condition (10) in Problem 13 if and only if there is a vector $\boldsymbol{\chi} \in \mathbb{F}_{q^m}[x;\sigma]^{\varrho}$ with*

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{\chi} \end{bmatrix} \cdot \boldsymbol{A} = \mathbf{0}, \tag{20}$$

where $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{(\ell+1+\varrho)\times\varrho}$ is a matrix whose i-th column, for $i = 1, \dots, \varrho$, is of the form

$$\left[\begin{array}{c} \mathbf{0}_{a_i \times 1} \\ \hline 1 \\ R_{a_i+2}^{(i)} \\ \vdots \\ R_{\ell+1}^{(i)} \\ \hline \mathbf{0}_{(i-1)\times 1} \\ \hline G^{(i)} \\ \hline \mathbf{0}_{(\varrho-i)\times 1} \end{array}\right]$$

where, for all $i = 1, \dots, \varrho$ and $j = a_i + 2, \dots, \ell+1$,

$$G^{(i)} := \mathcal{M}^{\mathrm{op}}_{\left\langle U_{1,1}^{(i)}, \dots, U_{\nu_i,1}^{(i)} \right\rangle}$$
$$R_j^{(i)} := \mathcal{I}^{\mathrm{op}}_{\left\{\left(U_{\kappa,1}^{(i)}, U_{\kappa,j-a_i}^{(i)}\right)\right\}_{\kappa=1}^{\nu_i}}.$$

Proof. A vector $\boldsymbol{Q} = [Q_0, \dots, Q_\ell] \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1}$ satisfies Condition (10) in Problem 13 on all rows of $\boldsymbol{U}$ if and only if each sub-block $[Q_{a_i}, \dots, Q_\ell]$ satisfies (10) on the rows of $U^{(i)}$. Using $G^{(i)}$ and $R_j^{(i)}$ as above, we can rewrite this condition, restricted to $\boldsymbol{U}^{(i)}$, as

$$\begin{aligned}
&\sum_{j=a_i+1}^{\ell+1} Q_{j-1}\left(U^{(i)}_{\kappa,j-a_i}\right) = 0 \quad \forall \kappa = 1, \dots, \nu_i \qquad (21)\\
\Leftrightarrow\ & Q_{a_i}\left(U^{(i)}_{\kappa,1}\right) + \sum_{j=a_i+2}^{\ell+1} Q_{j-1}\left(R_j^{(i)}\left(U^{(i)}_{\kappa,1}\right)\right) = 0 \quad \forall \kappa\\
\Leftrightarrow\ & \left(Q_{a_i} + \sum_{j=a_i+2}^{\ell+1} Q_{j-1} R_j^{(i)}\right)\left(U^{(i)}_{\kappa,1}\right) = 0 \quad \forall \kappa\\
\Leftrightarrow\ & Q_{a_i} + \sum_{j=a_i+2}^{\ell+1} Q_{j-1} R_j^{(i)} \equiv 0 \mod_{\mathrm{r}} \underbrace{\mathcal{M}^{\mathrm{op}}_{\left\langle U^{(i)}_{1,1},\dots,U^{(i)}_{\nu_i,1}\right\rangle}}_{=G^{(i)}}\\
\Leftrightarrow\ & \exists\, \chi_i \in \mathbb{F}_{q^m}[x;\sigma] \,:\, Q_{a_i} + \sum_{j=a_i+2}^{\ell+1} Q_{j-1} R_j^{(i)} + \chi_i G^{(i)} = 0\\
\Leftrightarrow\ & \exists\, \chi_i \in \mathbb{F}_{q^m}[x;\sigma] \,:\, \begin{bmatrix} Q_{a_i} & \cdots & Q_\ell & \chi_i \end{bmatrix} \cdot \begin{bmatrix} 1 \\ R^{(i)}_{a_i+2} \\ \vdots \\ R^{(i)}_{\ell+1} \\ G^{(i)} \end{bmatrix} = 0. \qquad (22)
\end{aligned}$$

This is equivalent to (20) since the χ_i's are independent of each other, but the Q_j are the same for each i. □

Example 19. *We give two examples for the matrix $\boldsymbol{A}$ as in Lemma 18.*

For $\nu_1 = n$ and $a_1 = 0$ ($\varrho = 1$), the first column of the matrix $\boldsymbol{U}$ already consists of linearly independent elements. This is an important special case since it is always fulfilled for the Wachter-Zeh–Zeh decoder (interleaved Gabidulin codes, see Section IV-A1). In this case, $\boldsymbol{A}$ of Lemma 18 has the form

$$\boldsymbol{A} = \begin{bmatrix} 1 \\ R_2^{(1)} \\ R_2^{(1)} \\ \vdots \\ R_{\ell+1}^{(1)} \\ G^{(1)} \end{bmatrix} \in \mathbb{F}_{q^m}[x;\sigma]^{(\ell+2)\times 1}.$$

For $a_i = i-1$, $i = 1, \dots, \varrho$, the matrix $\boldsymbol{A}$ has the form

$$\boldsymbol{A} = \begin{bmatrix}
1 & & & & \\
R_2^{(1)} & 1 & & & \\
R_3^{(1)} & R_3^{(2)} & 1 & & \\
\vdots & \vdots & \ddots & \ddots & \\
R_\varrho^{(1)} & R_\varrho^{(2)} & R_\varrho^{(3)} & \cdots & 1 \\
R_{\varrho+1}^{(1)} & R_{\varrho+1}^{(2)} & R_{\varrho+1}^{(3)} & \cdots & R_{\varrho+1}^{(\varrho)} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
R_{\ell+1}^{(1)} & R_{\ell+1}^{(2)} & R_{\ell+1}^{(3)} & \cdots & R_{\ell+1}^{(\varrho)} \\
G^{(1)} & & & & \\
 & G^{(2)} & & & \\
 & & G^{(3)} & & \\
 & & & \ddots & \\
 & & & & G^{(\varrho)}
\end{bmatrix}. \qquad (23)$$

In general, $\boldsymbol{A}$ has a form as in (23), where we delete the j-th column and $(\ell+1+j)$-th row (and rename the superscript indices accordingly) if there is no i with $a_i = j-1$.

Remark 20. *All vectors $\boldsymbol{Q} = [Q_0, \ldots, Q_\ell] \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1}$ satisfying Condition (10) form a left $\mathbb{F}_{q^m}[x;\sigma]$-module (see also [21]). Lemma 18 states that this module is the left kernel of $\boldsymbol{A}$, restricted to the first $\ell+1$ coordinates. Furthermore, it is the intersection of the left kernels of the columns of the matrix $\boldsymbol{A}$, which for $i = 1, \ldots, \varrho$ are the modules consisting of all vectors that, when restricted to the first $\ell+1$ coordinates, satisfy (10) with respect an alternative matrix of interpolation points of the form $\left[\boldsymbol{0}_{\nu_i,a_i} \mid \boldsymbol{U}^{(i)}\right] \in \mathbb{F}_{q^m}[x;\sigma]^{\nu_i \times (\ell+1)}$.*

Lemma 21. *Let $\boldsymbol{A}$ be defined as in Lemma 18, $\boldsymbol{w} \in \mathbb{Z}_{\geq 0}$, $D \in \mathbb{Z}_{>0}$. For $w_{\min} := \min_{i=1,\ldots,\ell+1}\{w_i\}$, set $d := D - w_{\min} + n$ and*

$$\boldsymbol{s} := [w_1, \ldots, w_{\ell+1}, w_{\min}, \ldots, w_{\min}] \in \mathbb{Z}_{\geq 0}^{\ell+1+\varrho}.$$

Then, for $\boldsymbol{Q} \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1}$ and $\boldsymbol{\chi} \in \mathbb{F}_{q^m}[x;\sigma]^{\varrho}$, we have

$$\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} \boldsymbol{A} = \boldsymbol{0} \quad \text{and} \tag{24}$$

$$\operatorname{rdeg}_{\boldsymbol{w}} \boldsymbol{Q} < D \tag{25}$$

if and only if

$$\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} \boldsymbol{A} \equiv \boldsymbol{0} \bmod_{\mathrm{r}} x^d \quad \text{and} \tag{26}$$

$$\operatorname{rdeg}_{\boldsymbol{s}} \begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} < D. \tag{27}$$

Proof. Let $\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix}$ satisfy (24) and (25). Then, obviously (26) holds. It is left to show the degree constraint. We have for the entries of $\boldsymbol{\chi} = [\chi_1, \ldots, \chi_\varrho]$

$$\deg \chi_i \leq \max_{j=i,\ldots,\ell+1}\{\deg Q_{j-1}\} - 1$$

since we can rewrite (24) into

$$-\chi_i G^{(i)} = Q_{i-1} + \sum_{j=i+1}^{\ell+1} Q_{j-1} R_j^{(i)} \quad \forall\, i = 1, \ldots, \varrho.$$

Due to $\deg G^{(i)} = \nu_i$ and $\deg R_j^{(i)} \leq \nu_i - 1$, we get the claimed degree bound on the χ_i. Hence, we have

$$\operatorname{rdeg}_{\boldsymbol{s}} \begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} = \max\Big\{\operatorname{rdeg}_{\boldsymbol{w}} \boldsymbol{Q}, \underbrace{w_{\min} + \max_i\{\deg \chi_i\}}_{\leq \operatorname{rdeg}_{\boldsymbol{w}} \boldsymbol{Q}}\Big\} < D.$$

For the other direction, the degree bound is obvious. As for the equality, the i-th entry (for $i = 1, \ldots, \varrho$) of $\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} \boldsymbol{A}$ is $Q_{i-1} + \sum_{j=i+1}^{\ell+1} Q_{j-1} R_j^{(i)} + \chi_i G^{(i)}$, where

$$\begin{aligned}
\deg Q_{i-1} &\leq D - w_i - 1 < D - w_{\min} \leq d,\\
\deg\left(Q_{j-1} R_j^{(i)}\right) &\leq D - w_j + \nu_i - 2 < D - w_{\min} + n = d,\\
\deg\left(\chi_i G^{(i)}\right) &\leq D - w_{\min} - 1 + \nu_i < D - w_{\min} + n = d,
\end{aligned}$$

thus $\operatorname{rdeg}\left(\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} \boldsymbol{A}\right) < d$. Hence, we have not only $\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} \boldsymbol{A} \equiv \boldsymbol{0} \bmod_{\mathrm{r}} x^d$, but also $\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} \boldsymbol{A} = \boldsymbol{0}$. □

Lemmas 18 and 21 combined imply a strategy for finding a basis of all solutions of Problem 13: compute a left approximant basis of $\boldsymbol{A}$ (both as defined in Lemma 18) with respect to the shift vector $\boldsymbol{s}$ and order d (as defined in Lemma 21). This strategy is outlined in Algorithm 6 and we give its complexity in Theorem 22.

Theorem 22. *Algorithm 6 is correct. For the complexity, assume $D \in \Theta(n)$. If the first column of the input matrix $\boldsymbol{U}$ consists of $\mathbb{F}_q$-linearly independent elements, it can be implemented with complexity*

$$\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$$

operations in $\mathbb{F}_q$. Otherwise, it costs

$$\tilde{O}\big(\ell^\omega \mathcal{M}_{q,m}(n) + \ell m n^{\omega-1}\big)$$

operations in $\mathbb{F}_q$.

Proof. Correctness follows by Lemmas 18 and 21, and the fact that $\boldsymbol{B}$ is in $\boldsymbol{s}$-ordered weak Popov form. The latter property implies that the left span of the rows of $\boldsymbol{B}$ indexed by $i_1, \ldots, i_{\ell'}$ includes all vectors satisfying both (26) and (27). Furthermore, by Lemma 21 these rows are in the left kernel of $\boldsymbol{A}$ (hence, if the row is $\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{\chi}\end{bmatrix} \neq \boldsymbol{0}$ we have $\deg \boldsymbol{Q} > \deg \boldsymbol{\chi}$ due to

Algorithm 6: Fast Interpolation Algorithm

Input : Instance of Problem 13: $\ell, n, D \in \mathbb{Z}_{>0}$, shift vector $\boldsymbol{w} \in \mathbb{Z}_{\geq 0}^{\ell+1}$, and $\boldsymbol{U} = [U_{i,j}] \in \mathbb{F}_{q^m}^{n \times (\ell+1)}$ with $\mathbb{F}_q$-linearly independent rows.

Output: If it exists, a solution of Problem 13. Otherwise, "no solution".

1 **if** *elements in first column of* $\boldsymbol{U}$ *are* $\mathbb{F}_q$*-lin. ind.* **then**
2 $\quad \boldsymbol{U}^{(1)} \leftarrow \boldsymbol{U}$, $\varrho \leftarrow 1$, $\nu_1 \leftarrow 1$, $a_1 \leftarrow 0$ super
3 **else**
4 $\quad \boldsymbol{U}^{(i)} \in \mathbb{F}_{q^m}^{\nu_i \times (\ell+1-a_i)}$ for $i = 1, \ldots, \varrho \leftarrow$ compute as in Lemma 17
5 **for** $i = 1, \ldots, \varrho$ **do**
6 $\quad G^{(i)} \leftarrow \mathcal{M}^{\mathrm{op}}_{\langle U^{(i)}_{1,1}, \ldots, U^{(i)}_{\nu_i,1} \rangle}$
7 $\quad$ **for** $j = a_i + 2, \ldots, \ell + 1$ **do**
8 $\quad\quad R^{(i)}_j \leftarrow \mathcal{I}^{\mathrm{op}}_{\left\{\left(U^{(i)}_{\kappa,1}, U^{(i)}_{\kappa,j-a_i}\right)\right\}_{\kappa=1}^{\nu_i}}$
9 $\boldsymbol{A} \leftarrow$ set up matrix from the $G^{(i)}$ and $R^{(i)}_j$ as in Lemma 18
10 $w_{\min} \leftarrow \min_{i=1,\ldots,\ell+1}\{w_i\}$
11 $d \leftarrow D - w_{\min} + n$
12 $\boldsymbol{s} \leftarrow [w_1, \ldots, w_{\ell+1}, w_{\min}, \ldots, w_{\min}] \in \mathbb{Z}_{\geq 0}^{\ell+1+\varrho}$
13 $\boldsymbol{B} \leftarrow$ left $\boldsymbol{s}$-ordered weak-Popov approximant basis of $\boldsymbol{A}$ of order d // Algorithm 5 in Section III
14 $\{i_1, \ldots, i_{\ell'}\} \leftarrow$ indices of rows of $\boldsymbol{B}$ with $\boldsymbol{s}$-shifted row degree $< D$
15 **if** $\ell' > 0$ **then**
16 $\quad$ **for** $j = 1, \ldots, \ell'$ **do**
17 $\quad\quad \boldsymbol{Q}^{(j)} \leftarrow [B_{i_j,1}, \ldots, B_{i_j,\ell+1}]$
18 $\quad$ **return** $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell')}$
19 **else**
20 $\quad$ **return** *"no solution"*

$\deg R^{(i)}_j < \deg G^{(i)}$ for all j), and due to the choice of $\boldsymbol{s}$, the $\boldsymbol{s}$-pivots of the rows of $\boldsymbol{B}$ indexed by $i_1, \ldots, i_{\ell'}$ are in the first $\ell + 1$ positions. This means that the $\boldsymbol{Q}^{(i)}$ (the restrictions of these rows to the first $\ell + 1$ components) have distinct $\boldsymbol{w}$-pivots, and are linearly independent. Hence, $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(\ell')}$ are a solution of Problem 13.

Recall from Section II-D that the annihilator polynomials $G^{(i)}$ and interpolation polynomials $R^{(i)}_j$ can be computed in $\tilde{O}(\mathcal{M}_{q,m}(\nu_i))$ operations each. Computing all the polynomials $G^{(i)}$ and $R^{(i)}_j$ with $i = 1, \ldots, \varrho$ and $j = i+1, \ldots, \ell+1$ hence costs at most

$$\tilde{O}\left(\ell \sum_{i=1}^{\varrho} \mathcal{M}_{q,m}(\nu_i)\right) \subseteq \tilde{O}(\ell \mathcal{M}_{q,m}(n))$$

since $\sum_{i=1}^{\varrho} \nu_i = n$ and $\mathcal{M}_{q,m}(\cdot)$ is a convex function.

Checking whether the first column of $\boldsymbol{U}$ has $\mathbb{F}_q$-rank n can be done by computing the remainder annihilator polynomial $A := \mathcal{M}^{\mathrm{op}}_{\langle U_{1,1}, \ldots, U_{n,1} \rangle}$ of the entries. The $U_{i,1}$ are linearly independent if and only if $\deg A = n$. This check can be done in $\tilde{O}(\mathcal{M}_{q,m}(n))$ (cf. Section II-D). Only if the entries are linearly independent, we need to compute the matrices $\boldsymbol{U}^{(i)}$ in Line 4. This costs $O(\ell m n^{\omega-1})$ operations in $\mathbb{F}_q$ (cf. Lemma 17).

By definition of $G^{(i)}$ and $R^{(i)}_j$, we have $\deg \boldsymbol{A} \leq n$. Due to $d \leq D + n$, Line 13 costs $\tilde{O}(\ell^{\omega} \mathcal{M}_{q,m}(n))$ by Theorem 11 in Section III. □

Algorithm 6 can also be phrased in the language of row reduction of an interpolation module basis (cf. [21], [51]) instead of approximant bases computation. We show in Appendix C how to construct a suitable module basis using the tools developed in this section.

C. A New Algorithm for the Root-Finding Step

The following lemma relates Problem 14 to computing a right approximant basis.

Lemma 23. *Consider an instance of Problem 14, with* $\hat{k} := \max_i\{k^{(i)}\}$*, and choose*

$$\boldsymbol{A} := \begin{bmatrix} Q_0^{(1)} & Q_1^{(1)} & \dots & Q_\ell^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ Q_0^{(\ell')} & Q_1^{(\ell')} & \dots & Q_\ell^{(\ell')} \end{bmatrix} \in \mathbb{F}_{q^m}[x;\sigma]^{\ell' \times (\ell+1)} \tag{28}$$

$$\boldsymbol{s} := \begin{bmatrix} \hat{k} & \hat{k} - k^{(1)} + 1 & \dots & \hat{k} - k^{(\ell)} + 1 \end{bmatrix} \in \mathbb{Z}_{\geq 0}^{\ell+1} \tag{29}$$

$$d := \max_{i,j} \left\{ \deg Q_j^{(i)} \right\} + \hat{k}. \tag{30}$$

Let $\boldsymbol{B} \in \mathbb{F}_{q^m}[x;\sigma]^{(\ell+1)\times(\ell+1)}$ *be a right* $\boldsymbol{s}$*-ordered weak-Popov approximant basis of* $\boldsymbol{A}$ *of order* d*. Then, with* $\boldsymbol{t} = \operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B})$*, the root space* $\mathcal{R}$ *defined in (12) of Problem 14 satisfies*

$$\begin{aligned} \mathcal{R} = \{ \begin{bmatrix} f^{(1)}, \dots, f^{(\ell)} \end{bmatrix}^\top : [f^{(0)}, \dots, f^{(\ell)}]^\top = \boldsymbol{B}\boldsymbol{v}, \\ \boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^{(\ell+1)\times 1} \text{ with } \operatorname{cdeg}_{\boldsymbol{t}} \boldsymbol{v} \leq \hat{k} \text{ and } f^{(0)} = 1 \}. \end{aligned} \tag{31}$$

Proof. By Lemma 1 then for any $\boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^{(\ell+1)\times 1}$, we have $\operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}\boldsymbol{v}) = \max_{i=1,\dots,\ell+1}\{\deg(v_i) + t_i\} = \operatorname{cdeg}_{\boldsymbol{t}} \boldsymbol{v}$.

$\subseteq$: Note that $\mathcal{R}$ consists of those vectors of the right-kernel of $\boldsymbol{A}$ having $\boldsymbol{s}$-degree at most $\hat{k}$ and first element being 1. Any such kernel vector $\boldsymbol{f}$ of $\boldsymbol{A}$ is in the column space of $\boldsymbol{B}$ by definition of approximant basis, so let $\boldsymbol{v}$ be such that $\boldsymbol{f} = \boldsymbol{B}\boldsymbol{v}$. But then we have $\operatorname{cdeg}_{\boldsymbol{t}} \boldsymbol{v} = \operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}\boldsymbol{v}) \leq \hat{k}$.

$\supseteq$: Let $\boldsymbol{v} \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1}$ with $\operatorname{cdeg}_{\boldsymbol{t}}(\boldsymbol{v}) \leq \hat{k}$. Then $\operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}\boldsymbol{v}) \leq \hat{k}$, i.e. $\operatorname{cdeg}(\boldsymbol{B}\boldsymbol{v}) \leq \hat{k} - \min(\boldsymbol{s}) < \hat{k}$. Since $\boldsymbol{B}$ is an approximant basis of $\boldsymbol{A}$, then $\boldsymbol{A}\boldsymbol{B}\boldsymbol{v} \equiv 0 \bmod_{\mathrm{l}} x^d$. But $\operatorname{cdeg}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{v}) \leq \max_{i,j}(\deg Q_j^{(i)}) + \operatorname{cdeg}(\boldsymbol{B}\boldsymbol{v}) < d$, and hence we can conclude $\boldsymbol{A}\boldsymbol{B}\boldsymbol{v} = 0$. In other words, $\boldsymbol{B}\boldsymbol{v}$ is a right kernel vector of $\boldsymbol{A}$. Since it also has $\boldsymbol{s}$-degree at most $\hat{k}$, it must be in $\mathcal{R}$ as long as its first component is 1. □

Lemma 23 gives an implicit description of the root space $\mathcal{R}$. The following lemma shows how to explicitly compute a basis of the affine module from $\boldsymbol{B}$.

Lemma 24. *Let* $\boldsymbol{B}$ *and* $\boldsymbol{t} = \operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B})$ *be defined as in Lemma 23. Denote by* $[B_{0,i}, \dots, B_{\ell,i}]^\top$ *the* i*-th column of* $\boldsymbol{B}$*, for* $i = 1, \dots, \ell+1$*. Let* $\mathcal{J}$ *be the set of indices of columns of* $\boldsymbol{B}$ *which have* $\boldsymbol{s}$*-degree at most* $\hat{k}$*, i.e.* $\forall i \in \mathcal{J}$ *we have* $t_i \leq \hat{k}$*, and let* $\mathcal{I} \subseteq \mathcal{J}$ *be those indices where the first entry of the corresponding column of* $\boldsymbol{B}$ *is not zero.*

If $\mathcal{I} = \emptyset$*, then* $\mathcal{R} = \emptyset$*. Otherwise, choose some* $i^* \in \mathcal{I}$*, denote by* $i_1, \dots, i_\iota$ *the distinct elements of* $\mathcal{I} \setminus \{i^*\}$ *and by* $j_1, \dots, j_\tau$ *the distinct elements of* $\mathcal{J} \setminus \mathcal{I}$*, respectively. Define*

$$\boldsymbol{g}^* := \tfrac{1}{B_{0,i^*}} [B_{1,i^*}, \dots, B_{\ell,i^*}]^\top \tag{32}$$

$$\boldsymbol{g}^{(r)} := [B_{1,i_r}, \dots, B_{\ell,i_r}]^\top - \tfrac{B_{0,i_r}}{B_{0,i^*}} [B_{1,i^*}, \dots, B_{\ell,i^*}]^\top \tag{33}$$

for $r = 1, \dots, \iota$*. For* $\delta = \iota + \sum_{i=1}^{\tau}(\hat{k} - t_{j_i} + 1)$*, define the vectors* $\boldsymbol{g}^{(\iota+1)}, \dots, \boldsymbol{g}^{(\delta)} \in \mathbb{F}_{q^m}[x;\sigma]^\ell$ *as*

$$\boldsymbol{g}^{\left(\iota + \sum_{i'=1}^{i-1}(\hat{k} - t_{j_i} + 1) + j + 1\right)} = [B_{1,j_i}, \dots, B_{\ell,j_i}]^\top x^j,$$

where $i = 1, \dots, \tau$ *and* $j = 0, \dots, \hat{k} - t_{j_i}$*. Then,* $\boldsymbol{g}^{(1)}, \dots, \boldsymbol{g}^{(\delta)}$ *are right linearly independent over* $\mathbb{F}_{q^m}$ *and*

$$\mathcal{R} = \boldsymbol{g}^* + \langle \boldsymbol{g}^{(1)}, \dots, \boldsymbol{g}^{(\delta)} \rangle_{\mathbb{F}_{q^m},\mathrm{right}}, \tag{34}$$

where $\langle \cdot \rangle_{\mathbb{F}_{q^m},\mathrm{right}}$ *denotes the right* $\mathbb{F}_{q^m}$*-span.*

Proof. According to Lemma 23, the roots contained in $\mathcal{R}$ are obtained from linear combinations $[f^{(0)}, \dots, f^{(\ell)}] = \boldsymbol{B}\boldsymbol{v}$ of the columns of $\boldsymbol{B}$ such that $\operatorname{cdeg}_{\boldsymbol{t}} \boldsymbol{v} \leq \hat{k}$ and $f^{(0)} = 1$. The first condition, $\operatorname{cdeg}_{\boldsymbol{t}} \boldsymbol{v} \leq \hat{k}$, implies that

- $v_i = 0$ for all $i \notin \mathcal{J}$ (since $t_i > \hat{k}$ in this case),
- $v_i \in \mathbb{F}_{q^m}$ for all $i \in \mathcal{I}$ (since $t_i = \hat{k}$), and
- $\deg v_i \leq \hat{k} - t_i$ for all $i \in \mathcal{J} \setminus \mathcal{I}$ (we write $v_i = \sum_{j=0}^{\hat{k}-t_i} x^j \tilde{v}_{i,j}$ with $\tilde{v}_{i,j} \in \mathbb{F}_{q^m}$ below).

If $\mathcal{I} = 0$, we cannot have $f^{(0)} \neq 0$, hence, $\mathcal{R} = \emptyset$. Else, $f^{(0)} = 1$ is equivalent to $\sum_{i\in\mathcal{I}} B_{0,i} v_i = 1$. By the elementary operations on the columns indexed by $\mathcal{I}$ (see (32) and (33)), we obtain the submatrix

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ \boldsymbol{g}^* & \boldsymbol{g}^{(1)} & \dots & \boldsymbol{g}^{(\iota)} \end{bmatrix} \in \mathbb{F}_{q^m}[x;\sigma]^{(\ell+1)\times(\iota+1)}.$$

By combining all conditions, we have $[f^{(1)}, \dots, f^{(\ell)}]^\top \in \mathcal{R}$ if and only if

$$\begin{aligned}\begin{bmatrix} f^{(1)} \\ \vdots \\ f^{(\ell)} \end{bmatrix} &= \boldsymbol{g}^* + \sum_{r=1}^{\iota} \boldsymbol{g}^{(r)} v'_{i_r} + \sum_{i=1}^{\tau} \begin{bmatrix} B_{1,j_i} \\ \vdots \\ B_{\ell,j_i} \end{bmatrix} \sum_{j=0}^{\hat{k}-t_{j_i}} \tilde{v}_{j_i,j}, \\ &= \boldsymbol{g}^* + \sum_{r=1}^{\iota} \boldsymbol{g}^{(r)} v'_{i_r} + \sum_{i=1}^{\tau} \sum_{j=0}^{\hat{k}-t_{j_i}} \boldsymbol{g}^{\left(\iota+\sum_{i'=1}^{i-1}(\hat{k}-t_{j_i}+1)+j+1\right)} \tilde{v}_{j_i,j},\end{aligned} \tag{35}$$

with some $v'_{i_r}, \tilde{v}_{j_i,j} \in \mathbb{F}_{q^m}$ for all r, i, j. This proves (34).

Since $\boldsymbol{B}$ is in $\boldsymbol{s}$-ordered column weak Popov form, for each root $[f^{(1)}, \dots, f^{(\ell)}]^\top \in \mathcal{R}$, there is a unique $\boldsymbol{v}$ with the given properties and $[f^{(1)}, \dots, f^{(\ell)}]^\top = \boldsymbol{B}\boldsymbol{v}$. We obtain the coefficients $v'_{i_r}, \tilde{v}_{j_i,j} \in \mathbb{F}_{q^m}$, for r, i, j, of the right $\mathbb{F}_{q^m}$-linear combination in (35) by a bijective mapping from the vector $\boldsymbol{v}$. Hence, the linear combination in (35) is unique for any root and the right $\mathbb{F}_{q^m}$-linearly independence of the $\boldsymbol{g}^{(i)}$ follows. □

Lemmas 23 and 24 imply a root-finding algorithm based on computing a right approximant basis. We outline the procedure in Algorithm 7 and prove its correctness and complexity in the following theorem.

Algorithm 7: Fast Root-Finding Algorithm

Input : Instance of Problem 14: $\ell, n \in \mathbb{Z}_{>0}$, $\boldsymbol{k} \in \mathbb{Z}_{>0}^{\ell}$, and left $\mathbb{F}_{q^m}[x;\sigma]$-linearly independent vectors $\boldsymbol{Q}^{(1)}, \dots, \boldsymbol{Q}^{(\ell')} \in \mathbb{F}_{q^m}[x;\sigma]^{\ell+1} \setminus \{\boldsymbol{0}\}$ with $\deg \boldsymbol{Q}^{(i)} \le n$ for all i.

Output: Solution of Problem 14: if $\mathcal{R} \neq \emptyset$, an affine basis $\boldsymbol{g}^*, \boldsymbol{g}^{(1)}, \dots, \boldsymbol{g}^{(\delta)}$ of the affine right $\mathbb{F}_{q^m}$-module $\mathcal{R}$ as defined in (12), i.e.,

$$\mathcal{R} = \boldsymbol{g}^* + \langle \boldsymbol{g}^{(1)}, \dots, \boldsymbol{g}^{(\delta)} \rangle_{\mathbb{F}_{q^m},\text{right}}.$$

If $\mathcal{R} = \emptyset$, "no solution"

$\hat{k} \leftarrow \max_i\{k^{(i)}\}$
$\boldsymbol{A} \leftarrow$ as in (28)
$\boldsymbol{s} \leftarrow [\hat{k}, \hat{k} - k^{(1)} + 1, \dots, \hat{k} - k^{(\ell)} + 1]$
$d \leftarrow \max_{i,j} \left\{ \deg Q_j^{(i)} \right\} + \hat{k}$
$\boldsymbol{B} \leftarrow$ right $\boldsymbol{s}$-ordered weak-Popov approximant basis of $\boldsymbol{A}$ of order d // Algorithm 4 in Section III
if *$\boldsymbol{B}$ has a row of* $\operatorname{rdeg}_{\boldsymbol{s}} \le \hat{k}$ **then**
  Compute $\boldsymbol{g}^*, \boldsymbol{g}^{(1)}, \dots, \boldsymbol{g}^{(\delta)}$ as in Lemma 24
  return $\boldsymbol{g}^*, \boldsymbol{g}^{(1)}, \dots, \boldsymbol{g}^{(\delta)}$
else
  return *"no solution"*

Theorem 25. *Algorithm 7 is correct. For the complexity, assume* $\max_i k^{(i)} \in \Theta(n)$. *Then, Algorithm 7 has complexity*

$$\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$$

Proof. Correctness follows from Lemmas 23 and 24. Complexity-wise the heaviest step is the computation of the right approximant basis, which costs $\tilde{O}\big(\ell^\omega \mathcal{M}_{q,m}(n + \max_i k^{(i)})\big) \subseteq \tilde{O}(\ell^\omega \mathcal{M}_{q,m}(n))$ since by assumption on the degree of $\boldsymbol{Q}^{(i)}$ in Problem 14, we have $d \le n + \max_i k^{(i)} \in \Theta(n)$. Computing the affine basis as in Lemma 24 costs $O(\ell^2 \max_i k^{(i)}) \subseteq O(\ell^2 n)$ operations over $\mathbb{F}_{q^m}$. □

V. Fast Decoding of Sum-Rank-Metric Codes

In this section, we show how to speed up decoding of linearized Reed–Solomon codes in the sum-rank metric. This is achieved by proposing new, faster, algorithms for the two core computational problems of the Martínez-Peñas–Kschischang decoder [7], which in fact decodes a more general class of codes in a more general metric: skew Reed–Solomon codes in the skew metric. We first state these problems and remind how the decoder works in Section V-A. We then present our new algorithms for them in Sections V-B and V-C.

In this section, we only use the remainder evaluation (cf. Section II-C) of skew polynomials.

A. Computational Problems and their Relation to Decoding

To state the two computational problems, we need to first recall some notions related to the remainder evaluation of skew polynomials.

1) Preliminaries on Remainder Evaluation: The following notions were introduced in [77], [67], and we use the notation of [7]. Let $A \subseteq \mathbb{F}_{q^m}[x;\sigma]$, $\Omega \subseteq \mathbb{F}_{q^m}$, and $a \in \mathbb{F}_{q^m}$. The *zero set* of A is defined by $Z(A) := \{\alpha \in \mathbb{F}_{q^m} : f[\alpha] = 0 \,\forall\, f \in A\}$, and $I(\Omega) := \{f \in \mathbb{F}_{q^m}[x;\sigma] : f[\alpha] = 0 \,\forall\, \alpha \in \Omega\}$ denotes the *associated ideal* of Ω. The *P-closure* of Ω is defined by $\overline{\Omega} := Z(I(\Omega))$, and Ω is called *P-closed* if $\overline{\Omega} = \Omega$. A P-closure is always P-closed. The elements of $\mathbb{F}_{q^m} \setminus \overline{\Omega}$ are all said to be *P-independent from* Ω.

A set $\mathcal{B} \subset \mathbb{F}_{q^m}$ is said to be *P-independent* if any $b \in \mathcal{B}$ is P-independent from $\mathcal{B} \setminus \{b\}$. If $\mathcal{B}$ is P-independent and $\Omega := \overline{\mathcal{B}} \subseteq \mathbb{F}_{q^m}$, we say that $\mathcal{B}$ is a *P-basis of* Ω. Ω may have many P-bases but they all have the same number of elements, called the *P-rank* of Ω, denoted $\mathrm{Prk}(\Omega) = |\mathcal{B}|$.

For any $\mathcal{B} \subset \mathbb{F}_{q^m}$ then $I(\mathcal{B})$ is a left $\mathbb{F}_{q^m}[x;\sigma]$-ideal and hence principal, so there is a unique monic skew polynomial $\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}}$ of smallest degree that generates it. We call $\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}}$ the *remainder annihilator polynomial* of $\mathcal{B}$ and we have $\deg \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}} = \mathrm{Prk}(\overline{\mathcal{B}})$. In particular, $\deg \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}} = |\mathcal{B}|$ if and only if $\mathcal{B}$ is P-independent.

Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \in \mathbb{F}_{q^m}$ be P-independent[5]. For any $\boldsymbol{r} = (r_1, \ldots, r_n) \in \mathbb{F}_{q^m}$, there is a unique skew polynomial $\mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}} \in \mathbb{F}_{q^m}[x;\sigma]$ of degree less than n such that

$$\mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}}[\beta_i] = r_i \quad \forall\, i = 1, \ldots, n.$$

We call this the *remainder interpolation polynomial* of $\boldsymbol{r}$ on $\mathcal{B}$.

2) Computational Problems: The decoder in [7] is based on the following computational problems.

Problem 26 (Fast Remainder-Evaluation Operations)**.** *Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \subseteq \mathbb{F}_{q^m}$ be P-independent.*

*i) Compute $\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}}$ (*remainder annihilator polynomial*).*
*ii) Given $f \in \mathbb{F}_{q^m}[x;\sigma]$ with $\deg f \leq n$, compute $\big[f[\beta_1], \ldots, f[\beta_n]\big]$ (*multi-point remainder evaluation*).*
*iii) Given $\boldsymbol{r} \in \mathbb{F}_{q^m}$, compute $\mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}}$ (*remainder interpolation*).*

Problem 27 (2D Vector Remainder Interpolation)**.** *Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \subseteq \mathbb{F}_{q^m}$ be P-independent. Given $D \in \mathbb{Z}_{>0}$, $\boldsymbol{w} \in \mathbb{Z}^2_{\geq 0}$, and $\boldsymbol{r} \in \mathbb{F}_{q^m}$, compute a non-zero $[Q_0, Q_1] \in \mathbb{F}_{q^m}[x;\sigma]^2$ such that*

$$Q_0[\beta_i] + (Q_1 R)[b_i] = 0 \quad \forall\, i = 1, \ldots, n, \tag{36}$$

$$\mathrm{rdeg}_{\boldsymbol{w}} \begin{bmatrix} Q_0 & Q_1 \end{bmatrix} < D, \tag{37}$$

where $R := \mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}}$.

As for Problem 13 in Section IV, we assume $D \in \Theta(n)$ for the complexity analysis. This is the only case relevant for the decoding problem studied in the following. See Section VI-C in the conclusion for a discussion on the general case. The previously fastest algorithms to solve Problems 26 and 27 with $D \in \Theta(n)$ were presented in [7] both of which uses $O(n^2)$ operations in $\mathbb{F}_{q^m}$.

3) Decoding of Skew Reed–Solomon Codes: Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\}$ be P-independent. The *skew Reed–Solomon code (w.r.t. $\mathcal{B}$)* [33] of dimension $k < n$ is defined as

$$\mathcal{C}_{\mathrm{skew},\mathcal{B}} := \left\{ \left[f[\beta_1], \ldots, f[\beta_n] \right] : f \in \mathbb{F}_{q^m}[x;\sigma]_{<k} \right\}.$$

The codes are designed for the skew metric, which is defined as follows. The *skew weight (w.r.t. $\mathcal{B}$)* [32] is

$$\begin{aligned} \mathrm{wt}_{\mathcal{B}} : \mathbb{F}_{q^m}^n &\to \mathbb{Z}_{\geq 0} \\ \boldsymbol{y} = [y_1, \ldots, y_n] &\mapsto n - \mathrm{Prk}\left(Z\left(\mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}}\right)\right). \end{aligned}$$

The *skew distance (w.r.t. $\mathcal{B}$)* is defined by $\mathrm{d}_{\mathcal{B}}(\boldsymbol{y}_1, \boldsymbol{y}_2) := \mathrm{wt}_{\mathcal{B}}(\boldsymbol{y}_1 - \boldsymbol{y}_2)$ for any $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{F}_{q^m}^n$. A skew Reed–Solomon code has minimum distance $d = n - k + 1$ w.r.t. the skew metric.

The skew metric is related to the sum-rank metric (see Theorem 28 below), which is defined as follows. As in Section IV, we define the ($\mathbb{F}_q$) rank weight of a row vector in $\mathbb{F}_{q^m}^{1 \times n'}$ as the $\mathbb{F}_q$-rank of the $m \times n'$ matrix over $\mathbb{F}_q$ obtained by column-wise expanding each entry of the vector in a basis of $\mathbb{F}_{q^m}$. For $\boldsymbol{n} = [n_1, \ldots, n_\ell]$ with $n_i \in \mathbb{Z}_{>0}$ and $\sum_{i=1}^{\ell} n_i = n$, the *sum-rank weight* (w.r.t. $\boldsymbol{n}$) on $\mathbb{F}_{q^m}^n$ [26] is defined as

$$\begin{aligned} \mathrm{wt}_{\mathrm{SR},\boldsymbol{n}} : \mathbb{F}_{q^m}^n &\to \mathbb{Z}_{\geq 0}, \\ \boldsymbol{c} = \left[\boldsymbol{c}^{(1)} \mid \cdots \mid \boldsymbol{c}^{(\ell)} \right] &\mapsto \sum_{i=1}^{\ell} \mathrm{wt}_{\mathrm{R}}\big(\boldsymbol{c}^{(i)}\big), \end{aligned}$$

where we divide $\boldsymbol{c}$ into subblocks $\boldsymbol{c}^{(i)} \in \mathbb{F}_{q^m}^{n_i}$. The *sum-rank distance* of $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_{q^m}^n$ is $\mathrm{d}_{\mathrm{SR},\boldsymbol{n}}(\boldsymbol{a}, \boldsymbol{b}) := \mathrm{wt}_{\mathrm{SR},\boldsymbol{n}}(\boldsymbol{a} - \boldsymbol{b})$.

[5]Here and in the sequel, we slightly abuse notation and take this to mean $\mathcal{B}$ is an ordered set and that the β_i are distinct.

Theorem 28 ([77], [67], [32], [7]). *Let $\boldsymbol{n} = [n_1, \ldots, n_\ell]$ with $n_i \in \mathbb{Z}_{>0}$ and $\sum_{i=1}^{\ell} n_i = n$, and let $m \in \mathbb{Z}_{>0}$ with $m \geq \max_i\{n_i\}$ and $\ell < q$ with q a prime power. Then there is a P-independent set $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \subset \mathbb{F}_{q^m}$ and non-zero field elements $\boldsymbol{v} = [v_1, \ldots, v_n] \in (\mathbb{F}_{q^m}^*)^n$ such that*

$$\begin{aligned} \varphi_{\mathcal{B},\boldsymbol{v}} \;:\; \left(\mathbb{F}_{q^m}^n, \mathrm{d}_{\mathcal{B}}\right) &\to \left(\mathbb{F}_{q^m}^n, \mathrm{d}_{\mathrm{SR},\boldsymbol{n}}\right), \\ \boldsymbol{c} = [c_1, \ldots, c_n] &\mapsto [c_1 v_1, \ldots, c_n v_n] \end{aligned}$$

is an isometry (i.e., bijective, distance-preserving mapping).

For a pair $\mathcal{B}$ and $\boldsymbol{v}$ as in Theorem 28, the linear code $\varphi_{\mathcal{B},\boldsymbol{v}}(\mathcal{C}_{\mathrm{skew},\mathcal{B}})$ is a *linearized Reed–Solomon code* as introduced in [32]. Since $\varphi_{\mathcal{B},\boldsymbol{v}}$ is an isometry, such a code has minimum sum-rank distance $n-k+1$ and is thus maximum distance separable in the sum-rank metric. Having precomputed $\boldsymbol{v}$, the isometry can be applied or reversed in only n multiplications in $\mathbb{F}_{q^m}$. Hence, any efficient decoder for skew Reed–Solomon codes in the skew metric is also an efficient decoder for linearized Reed–Solomon codes in the sum-rank metric. As skew Reed–Solomon codes are more general and can be described in skew polynomial language, we will only treat these codes in the following.

Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \in \mathbb{F}_{q^m}$ be P-independent. Let

$$\boldsymbol{r} = (r_1, \ldots, r_n) = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_{q^m}^n$$

such that $\boldsymbol{c}$ is a codeword of the skew Reed–Solomon code $\mathcal{C}_{\mathrm{skew},\mathcal{B}}[n,k]$ and $\boldsymbol{e}$ is an error of skew weight $\mathrm{wt}_{\mathcal{B}}(\boldsymbol{e})$. The Martínez-Peñas–Kschischang decoder [7] finds a solution $[Q_0, Q_1] \in \mathbb{F}_{q^m}[x;\sigma]^2$ of Problem 27 with input $D = \lfloor \frac{n-k}{2} \rfloor + k - 1$, $\boldsymbol{w} = [0, k-1]$, and $\{(\beta_i, r_i)\}_{i=1}^n$. It was shown in [7, Proposition 4] that if the skew weight of the error $\boldsymbol{e}$ is at most $\mathrm{wt}_{\mathcal{B}}(\boldsymbol{e}) \leq \lfloor \frac{n-k}{2} \rfloor$, then any such solution satisfies $-Q_0 = Q_1 f$, where $f \in \mathbb{F}_{q^m}[x;\sigma]_{<k}$ is the unique skew polynomial (i.e., message polynomial) of degree less than k with $\boldsymbol{c} = [f[\beta_1], \ldots, f[\beta_n]]$. Hence, to finish decoding once $[Q_0, Q_1]$ is obtained, we simply need to divide $-Q_0$ by Q_1 from the left and (multi-point) evaluate the resulting polynomial to obtain the original codeword $\boldsymbol{c}$.

Theorem 29. *Decoding a skew Reed–Solomon code $\mathcal{C}_{\mathrm{skew},\mathcal{B}}$ using the decoder in [7] has complexity $\tilde{O}(\mathcal{M}_{q,m}(n))$, if*

- *the 2D vector remainder interpolation is implemented using Algorithm 8 in Section V-C with input $D = \lfloor \frac{n-k}{2} \rfloor + k$, $\boldsymbol{w} = [0, k-1]$, and $\{(\beta_i, r_i)\}_{i=0}^n$;*
- *the univariate remainder interpolation and remainder annihilator computation inside Algorithm 8 are implemented using the algorithms implied by Theorems 30 and 32 in Section V-B;*
- *and, if the output should be the transmitted codeword instead of the message polynomial, the re-encoding is implemented using the fast multi-point evaluation algorithm implied by Theorem 31 in Section V-B.*

Decoding a linearized Reed–Solomon code can be done in the same cost through the isometry $\varphi_{\mathcal{B},\boldsymbol{v}}$.

Proof. The statement follows from [7, Proposition 4] (see summary above) and Theorems 30, 31, 32, and 35 (see next subsections). □

B. New Algorithms for Operations with Remainder Evaluation

We present fast algorithms to solve Problem 26: computing annihilators, multi-point evaluation, and remainder interpolation. The methods are similar to corresponding algorithms for the operator evaluation in [68, Lemma 3.3] (annihilator) and [70, Sections 3.4 and 3.5] (multi-point evaluation and interpolation), which are in turn non-commutative adaptions of well-known algorithms over ordinary polynomial rings (see, e.g., [78]).

Theorem 30 (Fast remainder annihilator polynomial computation). *Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\}$ be P-independent. Then $\mathcal{M}_{\mathcal{B}}^{\mathrm{rem}}$ can be computed in $\tilde{O}(\mathcal{M}_{q,m}(n))$ operations.*

Proof. Recall that the llcm of two skew polynomials $f, g \in \mathbb{F}_{q^m}[x;\sigma]$ is the unique monic skew polynomial $\mathrm{llcm}(f,g) \in \mathbb{F}_{q^m}[x;\sigma] \setminus \{0\}$ of smallest degree such that there are polynomials $\chi_1, \chi_2 \in \mathbb{F}_{q^m}[x;\sigma]$ with $\chi_1 f = \chi_2 g = \mathrm{llcm}(f,g)$. Note that we have $\deg \mathrm{llcm}(f,g) \leq \deg f + \deg g$.

Observe that if $\mathcal{B}_1, \mathcal{B}_2 \subset \mathbb{F}_{q^m}$ are disjoint, then $\mathrm{llcm}\left(\mathcal{M}_{\mathcal{B}_1}^{\mathrm{rem}}, \mathcal{M}_{\mathcal{B}_2}^{\mathrm{rem}}\right)$ is the least-degree monic polynomial in both the left ideal spanned by $\mathcal{M}_{\mathcal{B}_1}^{\mathrm{rem}}$ and by $\mathcal{M}_{\mathcal{B}_2}^{\mathrm{rem}}$, which must therefore be $\mathcal{M}_{\mathcal{B}_1 \cup \mathcal{B}_2}^{\mathrm{rem}}$. Furthermore, it is easy to see that $\mathcal{M}_{\{\beta\}}^{\mathrm{rem}} = x - \beta$ for any $\beta \in \mathbb{F}_{q^m}$. Recursively subdividing the initial $\mathcal{B}$ in disjoint subsets and structuring this this as a divide-&-conquer computation, the complexity $C(n)$ of computing $\mathcal{M}_{\mathcal{B}}^{\mathrm{rem}}$ as a function of n obeys $C(1) = O(1)$ and the recursion $C(n) = L(n) + 2C(\lceil n/2 \rceil), n > 1$, where $L(n)$ denotes the cost of computing the llcm of two skew polynomials of degree at most n. By [69, Theorem 3.2.7] $L(n) \subseteq \tilde{O}(\mathcal{M}_{q,m}(n))$, so by the master theorem, $C(n)$ is in the claimed complexity. □

Theorem 31 (Fast multi-point evaluation). *Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\}$ be P-independent and $f \in \mathbb{F}_{q^m}[x;\sigma]$ with $\deg f \leq n$. Then, $[f[\beta_1], \ldots, f[\beta_n]]$ can be computed in $\tilde{O}(\mathcal{M}_{q,m}(n))$ operations.*

Proof. Let $\mathcal{B} = \mathcal{B}_1 \sqcup \mathcal{B}_2$ be a partition of $\mathcal{B}$, and define

$$f_1 := f \bmod_{\mathrm{r}} \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_1},$$
$$f_2 := f \bmod_{\mathrm{r}} \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_2}.$$

Then for any $\beta \in \mathcal{B}$:

$$f[\beta] = \begin{cases} f_1[\beta], & \text{if } \beta \in \mathcal{B}_1 \\ f_2[\beta], & \text{if } \beta \in \mathcal{B}_2 \end{cases}.$$

Indeed for $j = 1, 2$, the polynomial $f - f_j$ is right-divisible by $\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_j}$ and hence $(f - f_j)[\beta] = 0$ for $\beta \in \mathcal{B}_j$.

Thus, if we split $\mathcal{B}$ in two parts of size $\leq n' := \lceil n/2 \rceil$, we can evaluate at each $\beta \in \mathcal{B}$ by computing two remainder annihilator polynomials of degree n', two right divisions of degree n, followed by two recursive multi-point evaluations of polynomials of degree at most n' in as many points. In the base case, we evaluate a polynomial of degree ≤ 1 at 1 point, which costs $O(1)$. By Theorem 30 and [69, Section 3.2.1] both the annihilator computations and divisions can be performed in $\tilde{O}(\mathcal{M}_{q,m}(n))$, and we obtain the claimed complexity using the master theorem. □

Theorem 32. *Let $\mathcal{B} = \{\beta_1, \ldots, \beta_n\}$ be P-independent and $\boldsymbol{r} \in \mathbb{F}_{q^m}^n$. Then the interpolation polynomial $\mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}} \in \mathbb{F}_{q^m}[x;\sigma]_{<n}$ can be computed in $\tilde{O}(\mathcal{M}_{q,m}(n))$ operations.*

Proof. Let $n' =\geq \lceil n/2 \rceil$, and $I = \{1, \ldots, n'\}$ and $J = \{n'+1, \ldots, n\}$ and set $\mathcal{B}_1 := \{\beta_i\}_{i \in I}$ and $\mathcal{B}_2 := \{\beta_i\}_{i \in J}$. We claim the identity:

$$\mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}} = \mathcal{I}^{\mathrm{rem}}_{\tilde{\mathcal{B}}_1,\tilde{\boldsymbol{r}}_1} \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_2} + \mathcal{I}^{\mathrm{rem}}_{\tilde{\mathcal{B}}_2,\tilde{\boldsymbol{r}}_2} \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_1},$$

where

$$\tilde{\mathcal{B}}_1 = \left\{ \frac{\sigma(\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_2}[\beta])\beta}{\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_2}[\beta]} \mid \beta \in \mathcal{B}_1 \right\}$$
$$\tilde{\mathcal{B}}_2 = \left\{ \frac{\sigma(\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_1}[\beta])\beta}{\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_1}[\beta]} \mid \beta \in \mathcal{B}_2 \right\}$$
$$\tilde{\boldsymbol{r}}_1 = \left(\frac{r_i}{\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_2}[\beta_i]} \right)_{i \in I}$$
$$\tilde{\boldsymbol{r}}_2 = \left(\frac{r_i}{\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_1}[\beta_i]} \right)_{i \in J}.$$

Indeed: the right-hand side clearly has degree less than n and remainder-evaluates to r_i at β_i for each $i \in \{1, \ldots, n\}$. Note that the P-independence of $\mathcal{B}$ implies $\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_j}[\beta_i] \neq 0$, so the $\tilde{\beta}_i$ and $\tilde{r}_i$ are well-defined. Furthermore, $\tilde{\mathcal{B}}_1$ is P-independent by the following argument. It follows from the product rule of remainder evaluation that the monic polynomial

$$\mathcal{M}^{\mathrm{rem}}_{\tilde{\mathcal{B}}_1} \cdot \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}_2}$$

vanishes on $\mathcal{B}$. Hence, it must be must be right-divisible by $\mathcal{M}^{\mathrm{rem}}_{\mathcal{B}}$, which has degree n by the P-independence of $\mathcal{B}$. This implies $\deg \mathcal{M}^{\mathrm{rem}}_{\tilde{\mathcal{B}}_1} \geq |\tilde{\mathcal{B}}_1|$ which implies the P-independence of $\tilde{\mathcal{B}}_1$. Mutadis mutandis, $\tilde{\mathcal{B}}_2$ is also P-independent, and the interpolation polynomials $\mathcal{I}^{\mathrm{rem}}_{\tilde{\mathcal{B}}_1,\tilde{\boldsymbol{r}}_1}$ and $\mathcal{I}^{\mathrm{rem}}_{\tilde{\mathcal{B}}_2,\tilde{\boldsymbol{r}}_2}$ are therefore well-defined.

Hence, we may compute $\mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}}$ by computing two remainder annihilator polynomials of size n', two multi-point evaluations of polynomials of degree at most n' on n' points, and recursively two interpolations on n' points. For the base case, we have $\mathcal{I}^{\mathrm{rem}}_{\beta,r} = (x - \beta) + r$ for any $\beta \in \mathbb{F}^*_{q^m}$ and $r \in \mathbb{F}_{q^m}$. By Theorems 30 and 31 and the master theorem, we obtain the desired complexity. □

C. A New Algorithm for the 2D Vector Interpolation Problem

The following statements reduce Problem 27 (2D vector remainder interpolation) to computing a left approximant basis. This will lead to a faster algorithm to solve the problem.

Lemma 33. *Consider an instance of Problem 27 and let $R := \mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}}$ and $G := \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}}$. Then, Condition (36) in Problem 27 is equivalent to*

$$Q_0 + Q_1 R \equiv 0 \mod_{\mathrm{r}} G. \tag{38}$$

Proof. First note that $Q_0 + Q_1 R \equiv 0 \bmod_{\mathrm{r}} G$ if and only if

$$\exists \chi \in \mathbb{F}_{q^m}[x;\sigma] \,:\, Q_0 + Q_1 R = \chi G.$$

Due to $G[b_i] = 0$, we have for all $i = 1, \ldots, n$

$$(Q_0 + Q_1 R)[b_i] = (\chi G)[b_i] = Q[b_i] + \underbrace{(\chi G)[b_i]}_{=0} = 0,$$

so (38) implies (36). For the other direction, we note that due to $(Q_0 + Q_1 R)]b_i] = 0$ for all i, we have $Q_0 + Q_1 R \in I(\mathcal{B})$. Since G generates the left ideal $I(\mathcal{B})$, there must be a polynomial $\chi \in \mathbb{F}_{q^m}[x;\sigma]$ with $Q_0 + Q_1 R = \chi G$. □

Lemma 34. *Consider an instance of Problem 27 and let* $R := \mathcal{I}^{\mathrm{rem}}_{\mathcal{B},\boldsymbol{r}} \in \mathbb{F}_{q^m}[x;\sigma]$ *and* $G := \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}}$. *Let* $\boldsymbol{s} = [s_1, s_2, s_3] := [w_1, w_2, \min\{w_1, w_2\}]$, *and* $d = D + n - \min\{w_1, w_2\}$, *as well as*

$$\boldsymbol{A} = \begin{bmatrix} 1 \\ R \\ G \end{bmatrix}.$$

Let $\boldsymbol{B}$ *be a left* $\boldsymbol{s}$*-ordered weak-Popov approximant basis of* $\boldsymbol{A}$ *of order* d. *Then Problem 27 has a solution if and only if* $\boldsymbol{B}$ *contains at least one row of* $\boldsymbol{s}$*-shifted degree at most* $D - 1$. *Furthermore, for any such row* $\boldsymbol{v} = [v_1, v_2, v_3]$, *then* $[Q_0, Q_1] := [v_1, v_2]$ *is a solution of Problem 27.*

Proof. Due to Lemma 33, Condition (36) in Problem 27 is equivalent to (38). It is easy to see that some $Q_0, Q_1 \in \mathbb{F}_{q^m}[x;\sigma]$ fulfill (36) if and only if there is a polynomial $\chi \in \mathbb{F}_{q^m}[x;\sigma]$ with

$$\begin{aligned} & Q_0 + Q_1 R + \chi G = 0 \\ \Leftrightarrow \quad & [Q_0, Q_1, \chi] \cdot \boldsymbol{A} = 0. \end{aligned}$$

Hence, the Q_0, Q_1 fulfilling (36) correspond directly to the vectors $[Q_0, Q_1, \chi]$ in the left kernel of the matrix $\boldsymbol{A}$. Furthermore, consider the shifted degree of such a Q_0, Q_1 which also satisfies the degree constraints of Problem 27:

$$\begin{aligned} \deg Q_0 + s_1 &< D, \\ \deg Q_1 + s_2 &< D, \\ \deg \chi + s_3 &= \deg(Q_0 + Q_1 R) + \min\{w_1, w_2\} - \deg G \\ &\leq \max\{\deg Q_0, \deg Q_1 + n - 1\} - n + \min\{w_1, w_2\} < D\ . \end{aligned}$$

In other words, $\mathrm{rdeg}_{\boldsymbol{s}}[Q_0, Q_1, \chi] < D$. Any vector $\boldsymbol{v} = [v_1, v_2, v_3] \in \mathbb{F}_{q^m}[x;\sigma]^3$ with $\mathrm{rdeg}_{\boldsymbol{s}} \boldsymbol{v} < D$ fulfills

$$\deg(\boldsymbol{v} \cdot \boldsymbol{A}) < D + n - \min\{w_1, w_2\},$$

so by the choice of d, any vector of this shifted degree is a left approximant of $\boldsymbol{A}$ of order d if and only if it is in the left kernel of $\boldsymbol{A}$.

Hence, the solutions of Problem 27 are exactly the first two entries of all non-zero left approximants of $\boldsymbol{A}$ of order d with $\boldsymbol{s}$-shifted degree at most $D - 1$. Since the rows of $\boldsymbol{B}$ are left approximants, any row of sufficiently small shifted degree is a solution of the problem. Moreover, the problem has a solution if and only if the row space of $\boldsymbol{B}$ contains a row of sufficiently small $\boldsymbol{s}$-shifted degree. Since $\boldsymbol{B}$ is in $\boldsymbol{s}$-shifted weak Popov form, one of its rows has minimal $\boldsymbol{s}$-shifted degree among all vectors of the row space, i.e., at most $D - 1$ if and only if the problem has a solution. □

Lemma 34 implies an algorithm to solve Problem 27, which we outline in Algorithm 8. We summarize its complexity in Theorem 35 below.

Algorithm 8: Fast 2D Vector Remainder Interpolation

Input : Instance of Problem 27: $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \subset \mathbb{F}_{q^m}$ and P-independent, $D \in \mathbb{Z}_{>0}$, $\boldsymbol{w} = [w_1, w_2] \in \mathbb{Z}^2_{\geq 0}$, and $\boldsymbol{r} \in \mathbb{F}^n_{q^m}$.

Output: Solution $[Q_0, Q_1] \in \mathbb{F}_{q^m}[x;\sigma]^2 \setminus \{\boldsymbol{0}\}$ if the problem has a solution, "no solution" otherwise.

$G \leftarrow \mathcal{M}^{\mathrm{rem}}_{\mathcal{B}}$ $R \leftarrow \mathcal{I}^{\mathrm{rem}}_{\{(\beta_i, r_i)\}_{i=1}^n}$ $\boldsymbol{s} \leftarrow [w_1, w_2, \min\{w_1, w_2\}]$

$d \leftarrow D + n - \min\{w_1, w_2\}$

$\boldsymbol{A} \leftarrow \begin{bmatrix} 1 \\ R \\ G \end{bmatrix}$

$\boldsymbol{B} \leftarrow$ left $\boldsymbol{s}$-ordered weak-Popov approximant basis of $\boldsymbol{A}$ of order d // Algorithm 5 in Section III

if $\boldsymbol{B}$ *has a row* $\boldsymbol{v} = [Q_0, Q_1, \chi]$ *of* $\mathrm{rdeg}_{\boldsymbol{s}} \boldsymbol{v} < D$ **then**

  return $[Q_0, Q_1]$

else

  return *"no solution"*

Theorem 35. *Algorithm 8 is correct. Assuming $D \in \Theta(n)$, it has complexity*

$$\tilde{O}(\mathcal{M}_{q,m}(n))$$

operations in $\mathbb{F}_{q^m}$.

Proof. Correctness follows directly from Lemma 34.

Setting up the matrix $\boldsymbol{A}$ consists of computing a remainder annihilator polynomial of degree n and an interpolation polynomial of degree $< n$. Both operations can be done in $\tilde{O}(\mathcal{M}_{q,m}(n))$ using Theorem 30 and 32, respectively. The approximant basis can be computed in $\tilde{O}(\mathcal{M}_{q,m}(\max\{D, n\})) \subseteq \tilde{O}(\mathcal{M}_{q,m}(n))$ operations using Algorithm 4 in Section III. □

VI. Conclusion

A. Summary

We have presented new algorithms for the underlying computational problems of three different decoders: interpolation-based decoding of interleaved Gabidulin codes in the rank metric, interpolation-based decoding of lifted interleaved Gabidulin codes in the subspace metric, and decoding of linearized/skew Reed–Solomon codes in the sum-rank/skew metric. Most of these computational problems were shown to be reducible to computing a left or right approximant basis over skew polynomial rings.

For all considered computational problems, hence also all considered decoders, we obtain an improvement in the dependence of the main parameter of a problem, say n, of the (soft-O) asymptotic complexity bound from a quadratic (or larger) dependence n^2 over $\mathbb{F}_{q^m}$ to the cost $\mathcal{M}_{q,m}(n)$ of multiplying two skew polynomials of degree at most n. Since the latter is sub-quadratic in n (at least $\mathcal{M}_{q,m}(n) \in O(n^{1.69})$, cf. Section II-D), we obtain significant speed-ups for all algorithms. See Tables I and II in the introduction for a detailed summary.

On the level of decoders, in the subspace- and sum-rank-metric cases we obtain faster decoding algorithms than previously known, while in the rank-metric case, we match the fastest state-of-the-art [4] for decoding interleaved Gabidulin codes with a different decoding method.

B. Further Applications

Some of the studied computational problems (cf. Table II in the introduction) have further applications beyond the scope of this paper, which we briefly summarize in the following. Since we have obtained faster algorithms to solve these problems, this might also influence these applications.

The vector (operator) interpolation (Problem 13) also corresponds to the interpolation steps in the decoding algorithms for Mahdavifar–Vardy [79], folded Gabidulin [80], and virtual interleaved Gabidulin [81] codes. Hence, Algorithm 6 immediate speeds up the interpolation steps of these decoders. Note that root finding in these algorithms is not an instance of the vector root-finding problem (Problem 14), hence further work is necessary to improve the overall complexity of these decoding algorithms.

Encoding in a linearized or skew Reed–Solomon code corresponds to a multi-point evaluation of a message polynomial at the evaluation points. Hence, Theorem 31 implies a faster encoder.

The maximally recoverable locally repairable (also called partial MDS) codes in [10] are defined via linearized Reed–Solomon codes. Repairing globally with these codes corresponds to erasure decoding of these codes and can be implemented by a skew polynomial remainder interpolation (part of Problem 26). Hence, the algorithm implied by Theorem 32 immediately speeds up the repair process of these codes.

C. Remarks on Generality

All definitions and statements in Section III (approximant bases), except for complexities, remain true when stated for skew polynomials over arbitrary finite Galois extensions $\mathbb{L}/\mathbb{K}$ instead of $\mathbb{F}_{q^m}/\mathbb{F}_q$ and automorphisms $\sigma \in \mathrm{Gal}(\mathbb{L}/\mathbb{K})$ with $\mathbb{K} = \mathbb{L}^\sigma$. The complexities are as stated if we in addition assume that there is a working basis of $\mathbb{L}/\mathbb{K}$ which allows to multiply, add, and apply σ to elements of $\mathbb{L}$ in $\tilde{O}([\mathbb{L} : \mathbb{K}])$ operations over $\mathbb{K}$ (this is the same assumption as in [68]).

The output of Algorithm 6 has slightly more structure than required by Problem 13 (vector operator interpolation problem in Section IV): the found $\mathbb{F}_{q^m}[x;\sigma]$-linearly independent vectors $\boldsymbol{Q}^{(1)}, \dots, \boldsymbol{Q}^{(\ell')}$ are reduced, i.e. the vector of $\boldsymbol{w}$-degrees is lexicographically minimal over all possible bases of $\mathcal{Q}$.

In Section IV, we assumed for the complexity analysis that the input parameters D and n of the vector interpolation problem (Problem 13) satisfy $D \in \Theta(n)$ since this is the only case relevant for the decoding problems considered here. It can be seen by adapting the proof of Theorem 22 that for general D and n, Algorithm 6 has complexity $\tilde{O}(\ell^\omega \mathcal{M}_{q,m}(D+n))$. Hence, for $D \ll n$ and $D \gg n$, the algorithm—as stated—is not faster than the one in [8], which has complexity $O(\ell^2 D n)$ over $\mathbb{F}_{q^m}$ in general. The details are out of the scope of this paper, but we briefly outline observations that we believe could lead to an improved cost of Algorithm 6 for these parameter ranges: If $n \ll D$, then the left kernel of $\boldsymbol{A}$ contains a basis of $\ell + 1$

elements, whose degree can be bounded only in n and $\boldsymbol{w}$. Hence, it appears possible to choose the order d of the sought approximant basis much smaller than $D+n$. The case $n \gg D$ may be improved by separating the interpolation constraints into $\approx n/D$ groups of D constraints each, and then chaining the minimal approximant basis computations while sifting out high-degree rows.

Analogously, we can improve the cost of solving Problem 27 (2D vector remainder interpolation in Section V) for $D \notin \Theta(n)$ by the same methods.

In Problem 14 (vector root-finding problem in Section IV), we assumed that $\max_i k^{(i)} \in \Theta(n)$. In general, Algorithm 7 has complexity $\tilde{O}\big(\ell^\omega \mathcal{M}_{q,m}(n + \max_i k^{(i)})\big)$. For $n \gg \max_i k^{(i)}$, this may be slower than the algorithms in [5], [6]. Again we believe Algorithm 7 could enjoy modifications similar to those outlined above for Algorithm 6 to handle these extremal parameter cases more efficiently.

D. Open Problems

The complexity bound of the new algorithm for the vector operator interpolation problem (Problem 13) has an extra term $O(\ell m n^{\omega-1})$ if the first components of the interpolation points are not $\mathbb{F}_q$-linearly independent (cf. Theorem 22). This is due to the fact that we first need to bring the interpolation point matrix into a specific form, which is algorithmically done by transforming an $n \times (\ell+1)m$ matrix over $\mathbb{F}_q$ into reduced row echelon form. Given the currently fastest skew-polynomial multiplication algorithms, the term $O(\ell m n^{\omega-1})$ is negligible compared to the $\ell^\omega \mathcal{M}_{q,m}(n)$ term. At this point, however, it is not known whether skew-polynomial multiplication *could* be sped up so this term is smallest for some parameters. It is known that square matrix multiplication and skew-polynomial multiplication are softly equivalent (i.e. $m^\omega \in \tilde{O}(\mathcal{M}_{q,m}(m))$ and $\mathcal{M}_{q,m}(m) \in \tilde{O}(m^\omega)$, cf. [68], [70]), and answering the above question seem to require relating square matrix multiplication with low-degree skew-polynomial multiplication.

Though we are not aware of an application, it is quite natural to generalize the 2D vector remainder interpolation problem (Problem 27) to larger dimensions, analog to the vector operator interpolation problem (Problem 13). If the first components of the evaluation points are P-independent, it appears to be straightforward to adapt the methods developed in Section IV-B (faster vector operator interpolation) to the $(\ell+1)$ dimensional vector remainder evaluation case. This corresponds to the special case that the first components of the interpolation points in Problem 13 are $\mathbb{F}_q$-linearly independent. It is not obvious how to solve the problem if the P-independence assumption is dropped.

References

[1] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde, "Fast Root Finding for Interpolation-Based Decoding of Interleaved Gabidulin Codes," in *IEEE Information Theory Workshop (ITW)*, 2019.

[2] C.-P. Jeannerod, V. Neiger, E. Schost, and G. Villard, "Computing Minimal Interpolation Bases," *Journal of Symbolic Computation*, vol. 83, pp. 272–314, Nov. 2017.

[3] B. Beckermann, H. Cheng, and G. Labahn, "Fraction-Free Row Reduction of Matrices of Skew Polynomials," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, Jul. 2002, pp. 8–15.

[4] V. Sidorenko and M. Bossert, "Fast Skew-Feedback Shift-Register Synthesis," *Designs, Codes and Cryptography*, vol. 70, no. 1-2, pp. 55–67, 2014.

[5] A. Wachter-Zeh and A. Zeh, "List and Unique Error-Erasure Decoding of Interleaved Gabidulin Codes with Interpolation Techniques," *Designs, Codes and Cryptography*, vol. 73, no. 2, pp. 547–570, 2014.

[6] H. Bartz and A. Wachter-Zeh, "Efficient List Decoding of Interleaved Subspace and Gabidulin Codes Using Gröbner Bases," *Advances in Mathematics of Communications*, vol. 12, no. 4, Nov. 2018.

[7] U. Martínez-Peñas and F. R. Kschischang, "Reliable and Secure Multishot Network Coding using Linearized Reed–Solomon Codes," *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 4785–4803, 2019.

[8] H. Xie, J. Lin, Z. Yan, and B. W. Suter, "Linearized Polynomial Interpolation and Its Applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 206–217, Jan. 2013.

[9] M. Alekhnovich, "Linear Diophantine Equations Over Polynomials and Soft Decoding of Reed–Solomon Codes," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2257–2265, Jul. 2005.

[10] U. Martínez-Peñas and F. R. Kschischang, "Universal and Dynamic Locally Repairable Codes with Maximal Recoverability via Sum-Rank Codes," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 7790–7805, 2019.

[11] P. Giorgi, C.-P. Jeannerod, and G. Villard, "On the Complexity of Polynomial Matrix Computations," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2003, pp. 135–142.

[12] P. Delsarte, "Bilinear Forms over a Finite Field with Applications to Coding Theory," *Journal of Combinatorial Theory, Series A*, vol. 25, no. 3, pp. 226–241, 1978.

[13] E. M. Gabidulin, "Theory of Codes with Maximum Rank Distance," *Problems of Information Transmission*, vol. 21, no. 1, pp. 3–16, 1985.

[14] R. M. Roth, "Maximum-Rank Array Codes and their Application to Crisscross Error Correction," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 328–336, 1991.

[15] D. Silva, F. R. Kschischang, and R. Koetter, "A Rank-Metric Approach to Error Control in Random Network Coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[16] V. Sidorenko and M. Bossert, "Decoding Interleaved Gabidulin Codes and Multisequence Linearized Shift-Register Synthesis," in *IEEE International Symposium on Information Theory (ISIT)*, 2010, pp. 1148–1152.

[17] C. Faure and P. Loidreau, "A New Public-Key Cryptosystem Based on the Problem of Reconstructing p-Polynomials," in *Coding and Cryptography*. Springer, 2006, pp. 304–315.

[18] R. Overbeck, "Public Key Cryptography Based on Coding Theory," Ph.D. dissertation, TU Darmstadt, 2007.

[19] P. Loidreau and R. Overbeck, "Decoding Rank Errors Beyond the Error Correcting Capability," in *International Workshop on Algebraic and Combinatorical Coding Theory (ACCT)*, 2006, pp. 186–190.

[20] V. Sidorenko, L. Jiang, and M. Bossert, "Skew-Feedback Shift-Register Synthesis and Decoding Interleaved Gabidulin Codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 621–632, 2011.

[21] S. Puchinger, J. Rosenkilde né Nielsen, W. Li, and V. Sidorenko, "Row Reduction Applied to Decoding of Rank-Metric and Subspace Codes," *Designs, Codes and Cryptography*, vol. 82, no. 1-2, pp. 389–409, 2017.

[22] R. Koetter and F. R. Kschischang, "Coding for Errors and Erasures in Random Network Coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.

[23] H. Wang, C. Xing, and R. Safavi-Naini, "Linear Authentication Codes: Bounds and Constructions," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 866–872, 2003.

[24] D. Silva, "Error Control for Network Coding," Ph.D. dissertation, University of Toronto, 2009.

[25] H. Bartz, M. Meier, and V. Sidorenko, "Improved Syndrome Decoding of Interleaved Subspace Codes," in *International ITG Conference on Systems, Communications and Coding (SCC)*, 2017.

[26] R. W. Nóbrega and B. F. Uchoa-Filho, "Multishot Codes for Network Coding Using Rank-Metric Codes," in *IEEE International Workshop on Wireless Network Coding*, 2010.

[27] A. Wachter, V. R. Sidorenko, M. Bossert, and V. V. Zyablov, "On (Partial) Unit Memory Codes Based on Gabidulin Codes," *Problems of Information Transmission*, vol. 47, no. 2, pp. 117–129, 2011.

[28] A. Wachter-Zeh and V. Sidorenko, "Rank Metric Convolutional Codes for Random Linear Network Coding," in *International Symposium on Network Coding (NetCod)*, 2012.

[29] A. Wachter-Zeh, M. Stinner, and V. Sidorenko, "Convolutional codes in rank metric with application to random network coding," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3199–3213, 2015.

[30] D. Napp, R. Pinto, J. Rosenthal, and P. Vettori, "MRD Rank Metric Convolutional Codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2766–2770.

[31] ——, "Faster Decoding of Rank Metric Convolutional Codes," in *International Symposium on Mathematical Theory of Networks and Systems*, 2018.

[32] U. Martínez-Peñas, "Skew and Linearized Reed–Solomon Codes and Maximum Sum Rank Distance Codes Over Any Division Ring," *Journal of Algebra*, vol. 504, pp. 587–612, 2018.

[33] D. Boucher and F. Ulmer, "Linear Codes Using Skew Polynomials with Automorphisms and Derivations," *Designs, Codes and Cryptography*, vol. 70, no. 3, pp. 405–431, 2014.

[34] P. Loidreau, "A Welch–Berlekamp Like Algorithm for Decoding Gabidulin Codes," in *Coding and Cryptography*. Springer, 2006, pp. 36–45.

[35] D. Boucher, "An Algorithm for Decoding Skew Reed–Solomon Codes with Respect to the Skew Metric," in *International Workshop on Coding and Cryptography (WCC)*, 2019.

[36] B. Beckermann and G. Labahn, "A Uniform Approach for Hermite Padé and Simultaneous Padé Approximants and Their Matrix-Type Generalizations," *Numerical Algorithms*, vol. 3, no. 1, pp. 45–54, 1992.

[37] M. V. Barel and A. Bultheel, "A General Module Theoretic Framework for Vector M-Padé and Matrix Rational Interpolation," *Numerical Algorithms*, vol. 3, no. 1, pp. 451–461, Dec. 1992.

[38] B. Beckermann and G. Labahn, "A Uniform Approach for the Fast Computation of Matrix-Type Padé Approximants," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 3, pp. 804–823, Jul. 1994.

[39] S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote, "Triangular-Basis Decompositions and Derandomization of Linear Algebra Algorithms Over," *Journal of Symbolic Computation*, vol. 47, no. 4, pp. 422–453, Apr. 2012.

[40] V. Neiger, "Fast Computation of Shifted Popov Forms of Polynomial Matrices via Systems of Modular Polynomial Equations," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, Jul. 2016.

[41] W. Zhou and G. Labahn, "Unimodular Completion of Polynomial Matrices," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2014, pp. 413–420.

[42] W. Zhou, G. Labahn, and A. Storjohann, "Computing Minimal Nullspace Bases," in *International Symposium on Symbolic and Algebraic Computation*, 2012, pp. 366–373.

[43] W. Zhou and G. Labahn, "Efficient Algorithms for Order Basis Computation," *Journal of Symbolic Computation*, vol. 47, no. 7, pp. 793–819, Jul. 2012.

[44] C.-P. Jeannerod, V. Neiger, E. Schost, and G. Villard, "Fast Computation of Minimal Interpolation Bases in Popov Form for Arbitrary Shifts," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2016.

[45] C.-P. Jeannerod, V. Neiger, and G. Villard, "Fast Computation of Approximant Bases in Canonical Form," *Journal of Symbolic Computation*, Jul. 2019.

[46] T. Kailath, *Linear Systems*. Prentice-Hall, 1980.

[47] V. Popov, "Some Properties of the Control Systems with Irreducible Matrix-Transfer Functions," in *Seminar on Differential Equations and Dynamical Systems, II*, 1970, pp. 169–180.

[48] T. Mulders and A. Storjohann, "On Lattice Reduction for Polynomial Matrices," *Journal of Symbolic Computation*, vol. 35, no. 4, pp. 377–401, 2003.

[49] J. S. R. Nielsen, "List Decoding of Algebraic Codes," Ph.D. dissertation, Technical University of Denmark, 2013.

[50] B. Beckermann and G. Labahn, "Fraction-Free Computation of Matrix Rational Interpolants and Matrix GCDs," *SIAM Journal on Matrix Analysis and Applications*, vol. 22, no. 1, pp. 114–144, Jan. 2000.

[51] S. Puchinger, S. Müelich, D. Mödinger, J. Rosenkilde, and M. Bossert, "Decoding Interleaved Gabidulin Codes Using Alekhnovich's Algorithm," *Electronic Notes in Discrete Mathematics*, vol. 57, pp. 175–180, 2017.

[52] S. Liu, F. Manganiello, and F. R. Kschischang, "Construction and Decoding of Generalized Skew-Evaluation Codes," in *IEEE Canadian Workshop on Information Theory (CWIT)*, 2015, pp. 9–13.

[53] S. Gao, "Normal Bases Over Finite Fields," Ph.D. dissertation, University of Waterloo, 1993.

[54] J.-M. Couveignes and R. Lercier, "Elliptic Periods for Finite Fields," *Finite Fields and Their Applications*, vol. 15, no. 1, pp. 1–22, 2009.

[55] F. Le Gall, "Powers of Tensors and Fast Matrix Multiplication," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2014, pp. 296–303.

[56] Ø. Ore, "Theory of Non-Commutative Polynomials," *Annals of Mathematics*, pp. 480–508, 1933.

[57] M. Bronstein and M. Petkovšek, "An Introduction to Pseudo-Linear Algebra," *Theoretical Computer Science*, vol. 157, no. 1, pp. 3–33, Apr. 1996.

[58] M. Kauers, "The holonomic toolkit," in *Computer Algebra in Quantum Field Theory*. Springer, 2013, pp. 119–144.

[59] Ø. Ore, "On a Special Class of Polynomials," *Transactions of the American Mathematical Society*, vol. 35, no. 3, pp. 559–584, Jul. 1933.

[60] S. D. Cohen and D. Hachenberger, "The Dynamics of Linearized Polynomials," *Proceedings of the Edinburgh Mathematical Society (Series 2)*, vol. 43, no. 01, pp. 113–128, 2000.

[61] R. J. Evans, J. Greene, H. Niederreiter *et al.*, "Linearized Polynomials and Permutation Polynomials of Finite Fields," *Michigan Mathematical Journal*, vol. 39, no. 3, pp. 405–413, 1992.

[62] B. Wu and Z. Liu, "Linearized Polynomials over Finite Fields Revisited," *Finite Fields and Their Applications*, vol. 22, pp. 79–100, 2013.

[63] P. L. Clark, "Non-Commutative Algebra (lecture notes)," 2012. [Online]. Available: http://math.uga.edu/~pete/noncommutativealgebra.pdf

[64] R. Lidl and H. Niederreiter, *Finite Fields*. Cambridge University Press, 1997, vol. 20.

[65] D. Augot, P. Loidreau, and G. Robert, "Generalized Gabidulin Codes Over Fields of Any Characteristic," *Designs, Codes and Cryptography*, vol. 86, no. 8, pp. 1807–1848, 2018.

[66] D. Silva and F. R. Kschischang, "Rank-Metric Codes for Priority Encoding Transmission with Network Coding," in *IEEE Canadian Workshop on Information Theory (CWIT)*, 2007, pp. 81–84.

[67] T.-Y. Lam and A. Leroy, "Vandermonde and Wronskian Matrices Over Division Rings," *Journal of Algebra*, vol. 119, no. 2, pp. 308–336, 1988.

[68] X. Caruso and J. Le Borgne, "Fast Multiplication for Skew Polynomials," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2017.
[69] ——, "A New Faster Algorithm for Factoring Skew Polynomials Over Finite Fields," *Journal of Symbolic Computation*, vol. 79, pp. 411–443, 2017.
[70] S. Puchinger and A. Wachter-Zeh, "Fast Operations on Linearized Polynomials and their Applications in Coding Theory," *Journal of Symbolic Computation*, vol. 89, pp. 194–215, 2018.
[71] S. Puchinger, "Construction and Decoding of Evaluation Codes in Hamming and Rank Metric," Ph.D. dissertation, Universität Ulm, 2018.
[72] Neiger, Vincent, "Bases of Relations in One or Several Variables: Fast Algorithms and Applications," Ph.D. dissertation, École Normale Supérieure de Lyon - University of Waterloo, 2016.
[73] B. Beckermann, H. Cheng, and G. Labahn, "Fraction-Free Row Reduction of Matrices of Ore Polynomials," *Journal of Symbolic Computation*, vol. 41, no. 5, pp. 513–543, 2006.
[74] P. Giorgi, C.-P. Jeannerod, and G. Villard, "On the Complexity of Polynomial Matrix Computations," in *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2003, pp. 135–142.
[75] A. Storjohann, "Algorithms for Matrix Canonical Forms," Ph.D. dissertation, ETH Zurich, 2000.
[76] H. Bartz, "Algebraic Decoding of Subspace and Rank-Metric Codes," Ph.D. dissertation, Technische Universität München, 2017.
[77] T.-Y. Lam, *A general theory of Vandermonde matrices*. Center for Pure and Applied Mathematics, University of California, Berkeley, 1985.
[78] J. Gathen and J. Gerhard, *Modern Computer Algebra*. Cambridge university press, 1999.
[79] H. Mahdavifar and A. Vardy, "Algebraic List-Decoding in Projective Space: Decoding with Multiplicities and Rank-Metric Codes," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 1085–1100, 2018.
[80] H. Bartz and V. Sidorenko, "Algebraic Decoding of Folded Gabidulin Codes," *Designs, Codes and Cryptography*, vol. 82, no. 1-2, pp. 449–467, 2017.
[81] V. Guruswami and C. Xing, "List Decoding Reed–Solomon, Algebraic-Geometric, and Gabidulin Subcodes Up to the Singleton Bound," in *ACM Symposium on the Theory of Computing*, 2013, pp. 843–852.

APPENDIX

A. Skew M-Basis Algorithm

In this section, we present right and left skew analogs of the M-basis algorithm [74, M-Basis]. The algorithms are asymptotically slower than the skew PM-basis algorithms presented in Section III-C2, but might be faster for small orders d since their hidden constant is smaller as they do not rely on asymptotically fast skew polynomial arithmetic (cf. Remark 12).

Algorithm 9: RightSkewMBasis

Input :
- positive integer $d \in \mathbb{Z}_{>0}$,
- matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ of degree $< d$,
- shifts $\boldsymbol{s} \in \mathbb{Z}^b$.

Output: $\boldsymbol{B} \in \mathsf{owPopovApprox}_{\mathsf{R}}(\boldsymbol{A}, \boldsymbol{s}, d)$

if *d=1* **then**
  return RightSkewBaseCase$(\boldsymbol{A}, \boldsymbol{s})$ // Algorithm 2
else
  $\boldsymbol{B}_1 \leftarrow \mathsf{RightSkewMBasis}\,(1, \boldsymbol{A} \operatorname{rem}_{\mathrm{l}} x, \boldsymbol{s})$
  $\boldsymbol{G} \leftarrow \left(x^{-1}\boldsymbol{A}\boldsymbol{B}_1\right) \operatorname{rem}_{\mathrm{l}} x^{d-1}$; $\boldsymbol{t} \leftarrow \operatorname{cdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$
  $\boldsymbol{B}_2 \leftarrow \mathsf{RightSkewMBasis}\,(d-1, \boldsymbol{G}, \boldsymbol{t})$
  return $\boldsymbol{B}_1\boldsymbol{B}_2$

Algorithm 10: LeftSkewMBasis

Input :
- positive integer $d \in \mathbb{Z}_{>0}$,
- matrix $\boldsymbol{A} \in \mathbb{F}_{q^m}[x;\sigma]^{a\times b}$ of degree $< d$,
- shifts $\boldsymbol{s} \in \mathbb{Z}^a$.

Output: $\boldsymbol{B} \in \mathsf{owPopovApprox}_{\mathsf{L}}(\boldsymbol{A}, \boldsymbol{s}, d)$

if *d=1* **then**
  return LeftSkewBaseCase$(\boldsymbol{A}, \boldsymbol{s})$ // Algorithm 3
else
  $\boldsymbol{B}_1 \leftarrow \mathsf{LeftSkewMBasis}\,(1, \boldsymbol{A} \operatorname{rem}_{\mathrm{r}} x^{\boldsymbol{s}})$
  $\boldsymbol{G} \leftarrow \left(\boldsymbol{B}_1\boldsymbol{A}x^{-1}\right) \operatorname{rem}_{\mathrm{r}} x^{d-1}$; $\boldsymbol{t} \leftarrow \operatorname{rdeg}_{\boldsymbol{s}}(\boldsymbol{B}_1)$
  $\boldsymbol{B}_2 \leftarrow \mathsf{RightSkewMBasis}\,(d-1, \boldsymbol{G}, \boldsymbol{t})$
  return $\boldsymbol{B}_2\boldsymbol{B}_1$

Theorem 36. *Algorithms 9 and 10 are correct. Algorithm 9 has complexity*

$$\tilde{O}\big(\max\{a,b\}b^{\omega-1}d^2\big)$$

and Algorithm 10 has complexity

$$\tilde{O}\big(a^{\omega-1}\max\{a,b\}d^2\big)$$

operations over $\mathbb{F}_{q^m}$.

Proof. Correctness follows from Lemma 10, as well as the correctness of the base cases (Theorem 7 for Algorithm 2 and Theorem 9 for Algorithm 3).

The base cases, Algorithm 2 for the left case and Algorithm 3 are called exactly d times. In the right case, Lines 5 and 7 are executed exactly $d-1$ times. Since $\boldsymbol{Q}$ has degree 0 and $\boldsymbol{B}_1$ has degree 1 (see proof of Theorem 7), the multiplication $x^{-1}\boldsymbol{Q}\boldsymbol{B}_1$ costs $O(\ell^\omega)$ operations in $\mathbb{F}_{q^m}$ and the multiplication $\boldsymbol{B}_1\boldsymbol{B}_2$ can be done in $O(\ell^\omega d)$. Overall, this costs $O(\ell^\omega d)$ over $\mathbb{F}_{q^m}$. The left case follows analogously. □

B. Examples

Here, we present some examples that are mentioned in the paper. Example 37 shows that we need to treat left and right approximant bases separately over skew polynomials (cf. Section III-B). This is different to the case of commutative polynomial rings.

Example 37. *Consider the field* $\mathbb{F}_{2^2}$ *(represented by* $\mathbb{F}_{2^2}=\mathbb{F}_2[b]/(b^2+1)$*), with* $\sigma=\cdot^2$*, and the following* 2×2 *matrix containing skew polynomials*

$$\boldsymbol{A}=\begin{bmatrix}(b+1)x^3+bx & x^3+bx^2+(b+1)x\\ (b+1)x^3+bx^2+x+b & x^3+x^2+1\end{bmatrix}$$

For $\boldsymbol{s}=[0,0]$ *and* $d=3$*, a left and a right* $\boldsymbol{s}$*-minimal approximant basis of* $\boldsymbol{A}$ *of order* d *are given as*

$$\boldsymbol{B}_{\text{left}}=\begin{bmatrix}x^2 & 0\\ bx+b & x\end{bmatrix}\in\mathbb{F}_{q^m}[x;\sigma]^{2\times 2} \quad \textit{and}$$
$$\boldsymbol{B}_{\text{right}}=\begin{bmatrix}x^2+(b+1)x & 1\\ x & x+b\end{bmatrix}\in\mathbb{F}_{q^m}[x;\sigma]^{2\times 2},$$

respectively. However, we have

$$\boldsymbol{A}^\top\boldsymbol{B}_{\text{left}}^\top \bmod_{\text{l}} x^3=\begin{bmatrix}0 & (b+1)x\\ 0 & x^2+(b+1)x\end{bmatrix},$$
$$\boldsymbol{B}_{\text{right}}^\top\boldsymbol{A}^\top \bmod_{\text{r}} x^3=\begin{bmatrix}0 & x^2+(b+1)x\\ x^2+(b+1)x & 0\end{bmatrix}.$$

Hence, in contrast to the ordinary polynomial ring $\mathbb{F}_{q^m}[x]$*, the matrix* $\boldsymbol{B}_{\text{left}}^\top$ *is not a right* $\boldsymbol{s}$*-minimal approximant basis of* $\boldsymbol{A}^\top$ *of order* d *and* $\boldsymbol{B}_{\text{right}}^\top$ *is not a left* $\boldsymbol{s}$*-minimal approximant basis of* $\boldsymbol{A}^\top$ *of order* d*.*

Example 38 shows that, in contrast to matrices of degree 0 and order 1, right approximant bases over skew polynomials cannot be in general computed from ones over ordinary polynomial rings using the mapping φ (cf. (2)). See Remark 8 in Section III-C for more details.

Example 38. *Consider the field* $\mathbb{F}_{2^2}$ *(represented by* $\mathbb{F}_{2^2}=\mathbb{F}_2[b]/(b^2+1)$*), with* $\sigma=\cdot^2$ *and the matrix*

$$\boldsymbol{A}=\begin{bmatrix}(b+1)x^2+(b+1) & bx^2+bx+(b+1)\\ x+b & x^2+bx+b\end{bmatrix}\in\mathbb{F}_{q^m}[x;\sigma]^{2\times 2}.$$

We want to compute an approximant basis of $\boldsymbol{A}$ *order* 2 *with respect to the shift vector* $\boldsymbol{s}=[0,0]$ *(i.e., unshifted). First, we compute*

$$\begin{aligned}\hat{\boldsymbol{A}}&=\varphi^{-1}(\boldsymbol{A})\\ &=\begin{bmatrix}(b+1)\,x^2+b+1 & bx^2+(b+1)\,x+b+1\\ x+b & x^2+(b+1)\,x+b\end{bmatrix}\in\mathbb{F}_{q^m}[x]^{2\times 2},\end{aligned}$$

and, using the PM-basis algorithm over $\mathbb{F}_{q^m}[x]$ *[74], [72], an* $\boldsymbol{s}$*-minimal approximant basis of order* 2 *of* $\hat{\boldsymbol{A}}$ *is,*

$$\hat{\boldsymbol{B}}=\begin{bmatrix}x+1 & x\\ 1 & x\end{bmatrix}\in\mathbb{F}_{q^m}[x]^{2\times 2}.$$

However, we have

$$\boldsymbol{A}\cdot\varphi\Big(\hat{\boldsymbol{B}}\Big) = \begin{bmatrix}(b+1)x^3+x^2+x & x^3+bx^2\\ x & x^3+(b+1)x^2\end{bmatrix} \equiv \begin{bmatrix}x & 0\\ x & 0\end{bmatrix} \bmod_{\mathrm{l}} x^2 \ ,$$

so the rows of $\varphi(\hat{\boldsymbol{B}})$ are not approximants of $\boldsymbol{A}$ of order 2.

C. Module Description of the Vector Operator Interpolation Problem

In this section we show how to find a basis for the left $\mathbb{F}_{q^m}[x;\sigma]$-module described by condition (10) in Problem 13. For a set of interpoaltion points $\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_n\}\in\mathbb{F}_{q^m}^s$ we define the corresponding left $\mathbb{F}_{q^m}[x;\sigma]$ module as

$$\mathfrak{M}(\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_n\}) := \{\boldsymbol{Q}\in\mathbb{F}_{q^m}[x;\sigma]^s : \boldsymbol{Q}(\boldsymbol{u}_i)=0, \quad \forall i=1,\ldots,n\}.$$

Note, that a basis for $\mathfrak{M}(\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{n_r}\})$ allows to solve Problem 13 using the row reduction methods from [21]. Note that this "pre-processing" step described in this section is necessary to generalize the special case of Problem 13 discussed in [21] (first column of $\boldsymbol{U}$ linearly independent).

As in Section IV-B, let $\boldsymbol{U}\in\mathbb{F}_{q^m}^{n_r\times(\ell+1)}$ be a basis for the received subspace $\mathcal{U}$ (i.e. we have $\langle\boldsymbol{U}\rangle_q=\mathcal{U}$). The matrix $\boldsymbol{U}$ contains the interpolation points $\boldsymbol{u}_i\in\mathbb{F}_{q^m}^{\ell+1}$ for $i=1,\ldots,n_r$ as rows. Recall, that a basis for the interpolation module cannot be set up using the ideas from [21, Lemma 5] since the first entries $u_{i,1}$ of the interpolation points $\boldsymbol{u}_i$ are not necessarily $\mathbb{F}_q$-linearly independent.

The following results lays the foundations for constructing a basis for the interpolation module $\mathfrak{M}(\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{n_r}\})$ recursively.

Consider a matrix $\boldsymbol{Z}\in\mathbb{F}_{q^m}^{n\times s}$ of the form

$$\boldsymbol{Z} = \left[\begin{array}{c|c}\multicolumn{2}{c}{\boldsymbol{Z}^{(1)}}\\ \hline \boldsymbol{0} & \boldsymbol{Z}^{(*)}\end{array}\right] \tag{39}$$

where $\boldsymbol{Z}^{(1)}\in\mathbb{F}_{q^m}^{\nu\times s}$ with $z_{1,1}^{(1)},\ldots,z_{\nu,1}^{(1)}$ being $\mathbb{F}_q$-linearly independent and $\boldsymbol{Z}^{(*)}\in\mathbb{F}_{q^m}^{(n-\nu)\times(s-1)}$. Denote by $\boldsymbol{z}_i$ and $\boldsymbol{z}_i^{(*)}$ the i-th row of $\boldsymbol{Z}$ and $\boldsymbol{Z}^{(*)}$, respectively.

Proposition 39. *If $\boldsymbol{L}\in\mathbb{F}_{q^m}[x;\sigma]^{(s-1)\times(s-1)}$ is a (lower-triangular) basis for $\mathfrak{M}(\{\boldsymbol{z}_1^{(*)},\ldots,\boldsymbol{z}_{n-\nu}^{(*)}\})\subseteq\mathbb{F}_{q^m}[x;\sigma]^{(s-1)}$, then the following matrix is a (lower-triangular) basis for $\mathfrak{M}(\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\})\subseteq\mathbb{F}_{q^m}[x;\sigma]^{s\times s}$:*

$$\boldsymbol{M} = \left[\begin{array}{c|c} G & \\ \hline R_1 & \\ \vdots & \boldsymbol{L}\\ R_{s-1} & \end{array}\right],$$

where

$$G \leftarrow \mathcal{M}^{\mathrm{op}}_{\langle z_{1,1}^{(1)},\ldots,z_{\nu,1}^{(1)}\rangle} \tag{40}$$

and each R_j is the interpolation skew polynomial given by:

$$R_j(z_{i,1}^{(1)}) = -\boldsymbol{L}_j(z_{i,2}^{(1)},\ldots,z_{i,s}^{(1)}) \ , \quad i=1,\ldots,\nu \ ,$$

where $\boldsymbol{L}_j$ is the j'th row of $\boldsymbol{L}$.

Proof. We first show that the rows of $\boldsymbol{M}$ are in $\mathfrak{M}(\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\})$. Clearly $G(z_{i,1})=0$ for all $i=1,\ldots,n$. For $1\le i\le\nu$, it is similarly obvious that $(R_j \mid \boldsymbol{L}_j)(\boldsymbol{z}_i)=0$, so remaining is only to show $(R_j \mid \boldsymbol{L}_j)(\boldsymbol{z}_i)=0$ for $i>\nu$. We have $(R_j \mid \boldsymbol{L}_j)(\boldsymbol{z}_i)=0 \iff \boldsymbol{L}_j\in\mathfrak{M}(\{\boldsymbol{z}_1^{(*)},\ldots,\boldsymbol{z}_{n-\nu}^{(*)}\})$ which is true.

To show that $\mathfrak{M}(\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\})$ is in the row span of $\boldsymbol{M}$, take any $\boldsymbol{Q}=(Q_1,\ldots,Q_s)\in\mathfrak{M}(\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\})$. We have that $(Q_2,\ldots,Q_s)\in\mathfrak{M}(\{\boldsymbol{z}_1^{(*)},\ldots,\boldsymbol{z}_{n-\nu}^{(*)}\})$, so there is a $\boldsymbol{q}\in\mathbb{F}_{q^m}[x;\sigma]^{s-1}$ such that $(Q_2,\ldots,Q_s)=\boldsymbol{q}\boldsymbol{L}$. Since the rows of $\boldsymbol{M}$ are in $\mathfrak{M}(\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\})$, so is the following vector:

$$\boldsymbol{Q}' = \boldsymbol{Q} - (0 \mid \boldsymbol{q})\boldsymbol{M} = (T,0,\ldots,0) \ .$$

Hence $T(z_{i,1})=0$ for $i=1,\ldots,\nu$, and so T must be right-divisible by G. □

Proposition 40. *Let $\boldsymbol{L} \in \mathbb{F}_{q^m}[x;\sigma]^{(s-1)\times(s-1)}$ be a (lower-triangular) basis for $\mathfrak{M}(\{\boldsymbol{z}_1^{(*)}, \dots, \boldsymbol{z}_{n-\nu}^{(*)}\}) \subseteq \mathbb{F}_{q^m}[x;\sigma]^{(s-1)}$, then the following matrix is a (lower-triangular) basis for $\mathfrak{M}(\{(0 \mid \boldsymbol{z}_1^{(*)}), \dots, (0 \mid \boldsymbol{z}_{n-\nu}^{(*)})\}) \subseteq \mathbb{F}_{q^m}[x;\sigma]^{s\times s}$:*

$$\boldsymbol{M} = \left[\begin{array}{c|c} 1 & \\ \hline 0 & \\ \vdots & \boldsymbol{L} \\ 0 & \end{array}\right] \tag{41}$$

Proof. We have that the first entries of the interpolation points are zero and thus not $\mathbb{F}_q$-linearly independent as in Proposition 39. However, the polynomials G and R_j from Proposition 39 are still well-defined. In particular, we have that $G \leftarrow \mathcal{M}^{\mathrm{op}}_{\langle 0,\dots,0\rangle} = 1$ and $R_j = 0$ since $R_j(0) = -\boldsymbol{L}_j(z_{i,1}^{(*)}, \dots, z_{i,s-1}^{(*)}) = 0$ for all $i = 1, \dots, n-\nu$ and $j = 1, \dots, s-1$. Using similar arguments as in the proof of Proposition 39 we have that the rows of $\boldsymbol{M}$ vanish on all interpolation points $(0 \mid \boldsymbol{z}_1^{(*)}), \dots, (0 \mid \boldsymbol{z}_{n-\nu}^{(*)})$ and form a basis for $\mathfrak{M}(\{(0 \mid \boldsymbol{z}_1), \dots, (0 \mid \boldsymbol{z}_n)\}) \subseteq \mathbb{F}_{q^m}[x;\sigma]^{s\times s}$. □

By applying the result of Proposition 39 and 40 recursively, we obtain Algorithm 11.

Remark 41. *Note, that if the entries $u_{1,1}^{(1)}, \dots, u_{n,1}^{(1)}$ are $\mathbb{F}_q$-linearly independent, the output of Algorithm 11 is a matrix $\boldsymbol{M}$ as given in [21, Lemma 5] for decoding interleaved Gabidulin codes. Hence, Algorithm 11 handles the general case for constructing a basis for the interpolation module.*

Algorithm 11: ModuleBasis$(\ell, \boldsymbol{U})$

Input : $\ell \in \mathbb{Z}_{>0}, \boldsymbol{U} \in \mathbb{F}_{q^m}^{n\times(\ell+1)}$ containing the interpolation points $\boldsymbol{u}_1, \dots, \boldsymbol{u}_n$ as rows.
Output: $\boldsymbol{M} \in \mathbb{F}_{q^m}[x;\sigma]^{(\ell+1)\times(\ell+1)}$, a lower-triangular basis of $\mathfrak{M}(\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_n\})$.

1 Compute the matrix $\boldsymbol{U}'$, ϱ, ν_i and a_i for all $i = 1, \dots, \varrho$ as in Lemma 17
2 $\boldsymbol{M} \leftarrow \boldsymbol{I}_{(\ell-a_\varrho)\times(\ell-a_\varrho)}$
3 $cnt \leftarrow \varrho$
4 **for** $i = 1, \dots, a_\varrho + 1$ **do**
5 $\boldsymbol{L} \leftarrow \boldsymbol{M}$
6 **if** $a_\varrho - i + 1 \neq a_{cnt}$ **then**
7 $G \leftarrow 1$
8 $R_j \leftarrow 0$ for all $j = 1, \dots, \ell - a_\varrho + i - 1$
9 **else**
10 $G \leftarrow \mathcal{M}^{\mathrm{op}}_{\langle u_{1,1}^{(cnt)}, \dots, u_{\nu_{cnt},1}^{(cnt)}\rangle}$
11 $R_j \leftarrow \mathcal{I}^{\mathrm{op}}_{\left\{\left(u_{\kappa,1}^{(cnt)}, \boldsymbol{L}_j(u_{\kappa,2}^{(cnt)}, \dots, u_{\kappa,\ell+1-a_{cnt}}^{(cnt)})\right)\right\}_{\kappa=1}^{\nu_{cnt}}}$ where $\boldsymbol{L}_j$ denotes the j-th row of $\boldsymbol{L}$ for $j = 1, \dots, \ell - a_{cnt}$
12 $$\boldsymbol{M} \leftarrow \left[\begin{array}{c|c} G & \\ \hline R_1 & \\ \vdots & \boldsymbol{L} \\ R_{\ell-a_{cnt}} & \end{array}\right].$$
  $cnt \leftarrow cnt - 1$
13 **return** $\boldsymbol{M}$

Theorem 42. *Algorithm 11 is correct. It has computational complexity $\tilde{O}(\ell^2 \mathcal{M}_{q,m}(n) + \ell m n^{\omega-1})$.*

Proof. The correctness of the algorithm follows by applying Proposition 39 and Proposition 40 recursively.

According to Lemma 17 the computation $\boldsymbol{U}'$ in Line 1 requires $O(\ell m n^{\omega-1})$ operations over $\mathbb{F}_q$. In each of the $a_\rho + 1 \in O(\ell)$ steps we need to construct the annihilator polynomial G, which requires $\tilde{O}(\mathcal{M}_{q,m}(\nu_i)) \in \tilde{O}(\mathcal{M}_{q,m}(n))$ operations. Line 11 corresponds to a multi-point evaluation of a row of $\boldsymbol{L}$ at at most n points, which requires $\tilde{O}(\ell \mathcal{M}_{q,m}(n))$ operations, and the construction of the interpolation polynomials which requires $\tilde{O}(\ell \mathcal{M}_{q,m}(n))$. Hence, the Algorithm requires at most $\tilde{O}(\ell^2 \mathcal{M}_{q,m}(n) + \ell m n^{\omega-1})$ operations.

□