Faster Algorithms for Multivariate Interpolation with Multiplicities and Simultaneous Polynomial Approximations

Muhammad F. I. Chowdhury, Claude-Pierre Jeannerod, Vincent Neiger, Eric Schost, Gilles Villard

▶ **To cite this version:**

Muhammad F. I. Chowdhury, Claude-Pierre Jeannerod, Vincent Neiger, Eric Schost, Gilles Villard. Faster Algorithms for Multivariate Interpolation with Multiplicities and Simultaneous Polynomial Approximations. 2014. hal-00941435v1

HAL Id: hal-00941435

https://inria.hal.science/hal-00941435v1

Preprint submitted on 3 Feb 2014 (v1), last revised 13 Feb 2015 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Faster Algorithms for Multivariate Interpolation with Multiplicities and Simultaneous Polynomial Approximations *

Muhammad F. I. Chowdhury†, Claude-Pierre Jeannerod‡, Vincent Neiger§, Éric Schost¶, Gilles Villard‖

February 4, 2014

Abstract

The interpolation step in the Guruswami-Sudan algorithm has attracted a lot of interest and it is now solved by many algorithms in the literature; they use either structured linear algebra or basis reduction for polynomial lattices. This problem of interpolation with multiplicities has been generalized to multivariate polynomials; in this context, to our knowledge only the approach based on polynomial lattices has been studied until now. Here, we reduce this interpolation problem to a problem of simultaneous polynomial approximations, which we solve using fast algorithms for structured linear algebra. This improves the best known complexity bounds for the interpolation step of the list-decoding of Reed-Solomon codes, Parvaresh-Vardy codes or folded Reed-Solomon codes. In the special case of Reed-Solomon codes, our approach has complexity $\mathcal{O}^{\sim}(\ell^{\omega-1}m^2n)$, where ℓ, m, n are the list size, the multiplicity and the number of sample points, and ω is the exponent of linear algebra; the speedup factor with comparison to the state of the art (Cohn and Heninger's algorithm) is ℓ/m, with $m \leqslant \ell$ in this context.

1 Introduction

Problems and results. In this paper, we consider a multivariate interpolation problem with multiplicities and degree constraints (Problem 1 below) which originates from coding theory. In what follows, $\mathbb{K}$ is our base field and, in the coding theory context, s, ℓ, m, n, b are respectively known as the *number of variables*, *list size*, *multiplicity*, *code length*, and as an *agreement parameter* (which is such that $n - b/m$ is an upper bound on the number of errors allowed on a received word). Furthermore, the s variables are associated with some *weights*

*The material in this paper was presented in part at the 10th Asian Symposium on Computer Mathematics (ASCM), Beijing, China, October 2012 and at SIAM Conference on Applied Algebraic Geometry, Fort Collins, Colorado, USA, August 2013.

†Computer Science Department, University of Western Ontario, London ON, Canada.

‡Inria, Laboratoire LIP (CNRS, ENS de Lyon, Inria, UCBL), Université de Lyon, France.

§ENS de Lyon, Laboratoire LIP (CNRS, ENS de Lyon, Inria, UCBL), Université de Lyon, France; Computer Science Department, University of Western Ontario, London ON, Canada.

¶Computer Science Department, University of Western Ontario, London ON, Canada.

‖CNRS, Laboratoire LIP (CNRS, ENS de Lyon, Inria, UCBL), Université de Lyon, France.

$k_1, \ldots, k_s$; in the coding theory context, all these weights are equal to a same value, say k, and $k+1$ corresponds to what is called the *message length* (or *dimension*) of the code.

We stress that here we do not address the issue of choosing the parameters s, ℓ, m with respect to $n, b, k_1, \ldots, k_s$, as is often done: in our context, these are all input parameters. Similarly, although we will mention them, we do not make some usual assumptions on these parameters; in particular, we do not make any assumption that ensures that our problem admits a solution: the algorithm will detect whether no solution exists.

Here and hereafter, $\mathbb{N}_{>0}$ is the set of positive integers, $\deg_X$ denotes the degree in the single variable X, and $\deg_{\boldsymbol{Y}}$ denotes the total degree with respect to the variables $Y_1, \ldots, Y_s$.

Problem 1. MULTIVARIATEINTERPOLATION

Input: s, ℓ, m, n, b in $\mathbb{N}_{>0}$, $(k_1, \ldots, k_s)$ in $\mathbb{N}_{>0}^s$ and points $\{(x_i, y_{i,1}, \ldots, y_{i,s})\}_{1 \leqslant i \leqslant n}$ in $\mathbb{K}^{s+1}$ with the x_i's pairwise distinct.

Output: a polynomial Q in $\mathbb{K}[X, Y_1, \ldots, Y_s]$ such that

(i) Q is nonzero,

(ii) $\deg_{\boldsymbol{Y}}(Q) \leqslant \ell$,

(iii) $\deg_X(Q(X, X^{k_1}Y_1, \ldots, X^{k_s}Y_s)) < b$,

(iv) for $1 \leqslant i \leqslant n$, $Q(x_i, y_{i,1}, \ldots, y_{i,s}) = 0$ with multiplicity at least m.

We call conditions (ii), (iii), and (iv) the *list-size* condition, the *weighted-degree* condition, and the *vanishing* condition, respectively. Furthermore, a point $(x_i, y_{i,1}, \ldots, y_{i,s})$ is a zero of Q of *multiplicity at least* m if the shifted polynomial $Q(X + x_i, Y_1 + y_{i,1}, \ldots, Y_s + y_{i,s})$ has no monomial of total degree less than m; in characteristic zero or larger than m, this is equivalent to require that all derivatives of Q of order up to $m-1$ vanish at $(x_i, y_{i,1}, \ldots, y_{i,s})$.

By linearizing condition (iv) under the assumption that conditions (ii) and (iii) are satisfied, it is easily seen that solving Problem 1 amounts to computing a nonzero solution to an $M \times N$ homogeneous linear system over $\mathbb{K}$. Here, the number M of equations derives from condition (iv) and thus depends on s, m, n, while the number N of unknowns derives from conditions (ii) and (iii) and thus depends on s, ℓ, b, and the k_i's. It is customary to assume $M < N$ in order to guarantee the existence of a nonzero solution; however, as said above, we do not make this assumption, since our algorithms do not require it.

Problem 1 is a generalization to s variables $Y_1, \ldots, Y_s$ of the interpolation step of list-decoding algorithms for Reed-Solomon codes: Sudan's algorithm [44] corresponds to $m = s = 1$, and Guruswami-Sudan's algorithm [22] to $s = 1$. Multivariate interpolation problems, with $s > 1$ and $k_1 = \cdots = k_s$, correspond for instance to Parvaresh-Vardy codes [35] or folded Reed-Solomon codes [21].

Our solution to Problem 1 relies on a reduction to a simultaneous approximation problem (Problem 2 below) which generalizes Padé and Hermite-Padé approximation.

Problem 2. SimultaneousPolynomialApproximations

Input: positive integers $\mu,\ \nu,\ M'_0, \ldots, M'_{\mu-1},\ N'_0, \ldots, N'_{\nu-1}$ and polynomial tuples $\{(P_i, F_{i,0}, \ldots, F_{i,\nu-1})\}_{0 \leqslant i < \mu}$ in $\mathbb{K}[X]^{\nu+1}$ such that for all i, P_i is monic of degree M'_i and $\deg(F_{i,j}) < M'_i$ for all j.

Output: polynomials $Q_0, \ldots, Q_{\nu-1}$ in $\mathbb{K}[X]$ satisfying the following conditions:

(a) the Q_j's are not all zero,

(b) for $0 \leqslant j < \nu$, $\deg(Q_j) < N'_j$,

(c) for $0 \leqslant i < \mu$, $\sum_{0 \leqslant j < \nu} Q_j F_{i,j} = 0 \bmod P_i$.

In this paper we present two algorithms to solve the latter problem. Both involve a linearization of the univariate equations (c) into a homogeneous linear system over $\mathbb{K}$; if we define

$$M' = \sum_{0 \leqslant i < \mu} M'_i \quad \text{and} \quad N' = \sum_{0 \leqslant j < \nu} N'_j,$$

then this system has M' equations in N' unknowns (note that as above, we will not assume that $M' < N'$).

Our two algorithms amount to reformulating this set of equations as a structured linear system, which we solve using the algorithm given by Bostan, Jeannerod, and Schost in [7]. The first approach, given in Section 4, follows the derivation of so-called extended key equations (EKE) presented in the case $s = 1, m = 1$ by Roth and Ruckenstein [38] and generalized to $s = 1, m \geqslant 1$ by Zeh, Gentner, and Augot [47]; the matrix of the system is mosaic-Hankel. In our second approach, presented in Section 5, the linear system is directly obtained from condition (c) without resorting to EKEs, and has Toeplitz-like structure.

Both points of view lead to the same complexity result, which says that Problem 2 can be solved in time quasi-linear in M', multiplied by a subquadratic term in $\rho = \max(\mu, \nu)$. In the following theorems, and the rest of this paper, the soft-O notation $\mathcal{O}^{\sim}(\ )$ indicates that we omit polylogarithmic terms. The exponent ω is so that we can multiply $n \times n$ matrices in $\mathcal{O}(n^\omega)$ ring operations on any ring, the best known bound being $\omega < 2.38$ [13, 43, 46, 17]. Finally, the function M is a *multiplication time* function for $\mathbb{K}[X]$: M is such that polynomials of degree at most d in $\mathbb{K}[X]$ can be multiplied in $\mathsf{M}(d)$ operations in $\mathbb{K}$, and such that M satisfies the super-linearity properties of [18, Ch. 8]. It follows from [10] that $\mathsf{M}(d)$ can be taken in $\mathcal{O}(d \log(d) \log\log(d)) \subset \mathcal{O}^{\sim}(d)$.

Theorem 1. *Let $\rho = \max(\mu, \nu)$. There exists a probabilistic algorithm that either computes a solution to Problem 2, or determines that none exists, using*

$$\mathcal{O}\big(\rho^{\omega-1} \mathsf{M}(M') \log(M')^2\big) \subset \mathcal{O}^{\sim}(\rho^{\omega-1} M')$$

operations in $\mathbb{K}$. Algorithm 2 in Section 4 and Algorithm 3 in Section 5 achieve this result. These algorithms both choose $\mathcal{O}(M')$ elements in $\mathbb{K}$; if these elements are chosen uniformly

at random in a set $S \subseteq \mathbb{K}$ *of cardinality at least* $6(M'+1)^2$*, then the probability of success is at least* $1/2$*.*

The probability analysis is a standard consequence of the Zippel-Schwartz lemma; as usual, the probability of success can be made arbitrarily close to one by increasing the size of S.

If the field $\mathbb{K}$ has fewer than $6(M'+1)^2$ elements, then a probability of success at least $1/2$ can still be achieved by using a suitable field extension $\mathbb{L}$, affecting only the logarithmic factor in the cost. Specifically, one can proceed in three steps. First, we take $\mathbb{L} = \mathbb{K}[X]/\langle f \rangle$ with $f \in \mathbb{K}[X]$ irreducible of degree $\Theta(\log(M'))$; such an f can be set up using an expected number of $\mathcal{O}^{\sim}((\log(M'))^2) \subset \mathcal{O}(M')$ operations in $\mathbb{K}$ [18, §14.9]. Then we solve Problem 2 over $\mathbb{L}$ by means of the algorithm of Theorem 1, thus using $\mathcal{O}\big(\rho^{\omega-1}\mathsf{M}(M')\log(M')^2 \cdot \mathsf{M}(\log(M'))\log\log(M')\big)$ operations in $\mathbb{K}$. Finally, from this solution over $\mathbb{L}$ one can deduce a solution over $\mathbb{K}$ for free. This last point comes from the fact that, as we shall see later in the paper, Problem 2 amounts to finding a nonzero vector u over $\mathbb{K}$ such that $Au = 0$ for some matrix A over $\mathbb{K}$: once we have obtained a solution $\overline{u}$ over $\mathbb{L}$, it thus suffices to rewrite it as $\overline{u} = \sum_{0 \leqslant i < \deg f} u_i X^i \neq 0$ and, noting that $Au_i = 0$ for all i, to return any nonzero u_i as a solution over $\mathbb{K}$.

We will use Theorem 1 to solve Problem 1 and obtain the following result.

Theorem 2. *Let*

$$\Gamma = \big\{(j_1, \ldots, j_s) \in \mathbb{N}^s \;\mid\; j_1 + \cdots + j_s \leqslant \ell \quad \textit{and} \quad j_1k_1 + \cdots + j_sk_s < b\big\},$$

and let $\varrho = \max\big(|\Gamma|, \binom{s+m-1}{s}\big)$ *and* $M = \binom{s+m}{s+1}n$*. There exists a probabilistic algorithm that either computes a solution to Problem 1, or determines that none exists, using*

$$\mathcal{O}\big(\varrho^{\omega-1}\mathsf{M}(M)\log(M)^2\big) \subset \mathcal{O}^{\sim}(\varrho^{\omega-1}M)$$

operations in $\mathbb{K}$*. This can be achieved using Algorithm 1 in Section 3 followed by either Algorithm 2 or Algorithm 3. These algorithms choose* $\mathcal{O}(M)$ *elements in* $\mathbb{K}$*; if these elements are chosen uniformly at random in a set* $S \subseteq \mathbb{K}$ *of cardinality at least* $6(M+1)^2$*, then the probability of success is at least* $1/2$*.*

If $|\mathbb{K}| < 6(M+1)^2$ then, as before, a probability of success at least $1/2$ can still be ensured via an extension of $\mathbb{K}$ of suitable degree d, up to a cost increase by a factor in $\mathcal{O}(\mathsf{M}(d)\log(d))$. Since the x_i in Problem 1 are assumed to be pairwise distinct, we have already $|\mathbb{K}| \geqslant n$ and thus we can take $d = \mathcal{O}(\log_n(M))$.

In order to understand the cost estimate in Theorem 2, we now briefly discuss it under the following usual assumptions on the input parameters:

$\mathbf{H_1}$: $m \leqslant \ell$,

$\mathbf{H_2}$: $k_1 = \cdots = k_s =: k$,

and, under $\mathbf{H_2}$,

$\mathbf{H_3}$: $\ell k < b$,

$\mathbf{H_4}$: $k < n$.

With regards to the assumption $\mathbf{H_1}$ we mention in Appendix A that the case $m \geqslant \ell$ can easily be reduced to the case $m = \ell$. The second assumption $\mathbf{H_2}$ states that $k := k_1 = \cdots = k_s$: this corresponds to the coding theory context, where $k+1$ is the message length. The third assumption $\mathbf{H_3}$ means that we do not take ℓ uselessly large: if $\ell k \geqslant b$, then the weighted-degree constraint implies that some of the coefficients $Q_{\boldsymbol{j}}$ are identically zero. Finally, assumption $\mathbf{H_4}$ is natural in the coding theory context: since $k+1$ is the message length, it must be at most n. To support this last assumption independently from any application context, we show in Appendix B that with the only assumption "$k_r \geqslant n$ for all $r \in \{1, \ldots, s\}$" on parameters for Problem 1, this problem has either a trivial solution or no solution at all.

Under the assumptions $\mathbf{H_2}$ and $\mathbf{H_3}$, the set Γ introduced in Theorem 2 reduces to $\{(j_1, \ldots, j_s) \in \mathbb{N}^s \,:\, j_1 + \cdots + j_s \leqslant \ell\}$, so that $|\Gamma| = \binom{s+\ell}{s}$. Thus, if we further assume $\mathbf{H_1}$, the parameter ϱ in that theorem has the form $\varrho = \binom{s+\ell}{s}$. Assume for simplicity that s is constant; then, ϱ and M grow respectively like ℓ^s and $m^{s+1}n$. As a particular case, we obtain the following complexity result, which applies to the interpolation step of Guruswami and Sudan's list-decoding algorithm:

Corollary 3. *Taking $s = 1$, if the parameters ℓ, m, n, b, and $k := k_1$ satisfy $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{H}_3$ and $\mathbf{H}_4$, then there exists a probabilistic algorithm that computes a solution to Problem 1 using*

$$\mathcal{O}\big(\ell^{\omega-1}\mathsf{M}(m^2 n)\log(mn)^2\big) \subset \mathcal{O}^{\sim}(\ell^{\omega-1}m^2 n)$$

operations in $\mathbb{K}$. This can be achieved using Algorithm 1 followed by either Algorithm 2 or Algorithm 3. These algorithms choose $\mathcal{O}(m^2 n)$ elements in $\mathbb{K}$; if these elements are chosen uniformly at random in a set $S \subseteq \mathbb{K}$ of cardinality at least $24m^4n^2$, then the probability of success is at least $1/2$.

If $|\mathbb{K}| < 24m^4n^2$ then extension degree $d = \mathcal{O}(\log_n(m))$ suffices, so that the cost above becomes $\mathcal{O}\big(\ell^{\omega-1}\mathsf{M}(m^2 n)\log(mn)^2 \cdot \mathsf{M}(d)\log(d)\big)$. In the applications to coding theory, we have $m = n^{\mathcal{O}(1)}$, so that $d = \mathcal{O}(1)$ and then the cost and probability of success in Corollary 3 hold for *any* field $\mathbb{K}$.

Notation. For simplicity, in the rest of this paper we will use boldface letters to denote s-tuples of objects: $\boldsymbol{Y} = (Y_1, \ldots, Y_s)$, $\boldsymbol{k} = (k_1, \ldots, k_s)$, etc. In the special case of s-tuples of integers, we also write $|\boldsymbol{k}| = k_1 + \cdots + k_s$. Concerning Problem 1, we will repeatedly use the master polynomial associated with the x_i's, defined as

$$G(X) = \prod_{i=1}^{n}(X - x_i), \tag{1}$$

as well as the s-tuple $\boldsymbol{R} = (R_1, \ldots, R_s) \in \mathbb{K}[X]^s$ of Lagrange interpolation polynomials, defined by the conditions

$$\deg(R_j) < n \quad \text{and} \quad R_j(x_i) = y_{i,j} \tag{2}$$

for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant s$.

Comparison with previous work. Regarding Problem 1, most previous results focus on the Guruswami-Sudan case $s = 1$ and some of them on the Sudan case $s = 1, m = 1$. We summarize these results in Table 1, in which we make assumptions $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$; in some cases [36, 1, 5, 12], the complexity was not stated quite exactly in our terms but the translation is straightforward.

In the second column of that table, we give the cost with respect to the interpolation parameters ℓ, m, n, assuming further $m = n^{\mathcal{O}(1)}$ and $\ell = n^{\mathcal{O}(1)}$. The most significant factor in the running time is its dependency with respect to n, with results being either cubic, quadratic, or quasi-linear. Then, under the assumption $\mathbf{H_1}$: $m \leqslant \ell$, the second most important parameter is ℓ, followed by m. In particular, our result in Corollary 3 compares favorably to the cost $\mathcal{O}^{\sim}(\ell^{\omega} mn)$ obtained by Cohn and Heninger [12], which was, to our knowledge, the best previous bound for this problem.

In the third column, we give the cost with respect to the Reed-Solomon code parameters n and k, using worst-case parameter choices that are made to ensure the existence of a solution: $m = \mathcal{O}(nk)$ and $\ell = \mathcal{O}(n^{3/2}k^{1/2})$ in the Guruswami-Sudan case [22] and $\ell = \mathcal{O}(n^{1/2}k^{-1/2})$ in the Sudan case [44]. With these parameter choices, our algorithms present a speedup $(n/k)^{1/2}$ over the algorithm in [12].

In the general case $s \geqslant 1$, the result in Theorem 2 improves as well on the best previously known bounds; we discuss those below, under the assumptions $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$.

Most previous algorithms rely on linear algebra, either over $\mathbb{K}$ or over $\mathbb{K}[X]$. Working over $\mathbb{K}$, a natural idea is to rely on cubic-time general linear system solvers, as in Sudan's and Guruswami-Sudan's original papers. Several papers also cast the problem in terms of Gröbner basis computation in $\mathbb{K}[X, Y]$, implicitly or explicitly: the incremental algorithms of [26, 32, 28] are particular cases of the Buchberger-Möller algorithm [29], while Alekhnovich's algorithm [1] is a divide-and-conquer change-of-order for bivariate ideals. In [16], the authors propose a generalization of the incremental algorithm for Newton interpolation to the multivariate case with multiplicities. For simple roots and under some genericity assumption on the points $\{(x_i, y_{i,1}, \ldots, y_{i,s})\}_{1 \leqslant i \leqslant n}$, this algorithm uses $O(n^2)$ operations to compute a polynomial Q which satisfies (i), (iii), (iv) with $m = 1$. However, the complexity analysis is not clear to us in the general case with multiple roots ($m \geqslant 1$).

Yet another line of work [38, 47] uses Feng-Tzeng's linear system solver [15], combined with a reformulation in terms of syndromes and key equations. We will use (and generalize to the case $s > 1$) some of these results in Section 4, but we will rely on the structured linear system solver of [7] in order to prove our main results. Prior to our work, Olshevsky and Shokrollahi also used structured linear algebra techniques [33], but it is unclear to us whether their encoding of the problem could lead to similar results as ours.

Table 1: Comparison of our costs with previous ones for $s = 1$

Sudan case ($m = 1$)		
Sudan [44]	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
Roth-Ruckenstein [38]	$\mathcal{O}(\ell n^2)$	$\mathcal{O}(n^{2+1/2}k^{-1/2})$
Olshevsky-Shokrollahi [33]	$\mathcal{O}(\ell n^2)$	$\mathcal{O}(n^{2+1/2}k^{-1/2})$
This paper	$\mathcal{O}(\ell^{\omega-1}\mathsf{M}(n)\log(n)^2)$	$\mathcal{O}^{\sim}(n^{\omega}k^{1/2-\omega/2})$
Guruswami-Sudan case ($m \geqslant 1$)		
Guruswami-Sudan [22]	$\mathcal{O}(m^6n^3)$	$\mathcal{O}(n^9k^6)$
Olshevsky-Shokrollahi [33]	$\mathcal{O}(\ell m^4n^2)$	$\mathcal{O}(n^{7+1/2}k^{4+1/2})$
Augot-Gentner-Zeh [47]	$\mathcal{O}(\ell m^4n^2)$	$\mathcal{O}(n^{7+1/2}k^{4+1/2})$
Kötter / McEliece [26, 28]	$\mathcal{O}(\ell m^4n^2)$	$\mathcal{O}(n^{7+1/2}k^{4+1/2})$
Reinhard [36]	$\mathcal{O}(\ell^3m^4n^2)$	$\mathcal{O}(n^{10+1/2}k^{5+1/2})$
Lee-O'Sullivan [27]	$\mathcal{O}(\ell^4mn^2)$	$\mathcal{O}(n^9k^3)$
Trifonov [45] (heuristic)	$\mathcal{O}(m^3n^2)$	$\mathcal{O}(n^5k^3)$
Alekhnovich [1]	$\mathcal{O}(\ell^4m^4\mathsf{M}(n)\log(n))$	$\mathcal{O}^{\sim}(n^{11}k^6)$
Beelen-Brander [4]	$\mathcal{O}(\ell^3\mathsf{M}(\ell mn)\log(n))$	$\mathcal{O}^{\sim}(n^8k^3)$
Bernstein [5]	$\mathcal{O}(\ell^{\omega}\mathsf{M}(\ell n)\log(n))$	$\mathcal{O}^{\sim}(n^{3\omega/2+5/2}k^{\omega/2+1/2})$
Cohn-Heninger [12]	$\mathcal{O}(\ell^{\omega}\mathsf{M}(mn)\log(n))$	$\mathcal{O}^{\sim}(n^{3\omega/2+2}k^{\omega/2+1})$
This paper	$\mathcal{O}\big(\ell^{\omega-1}\mathsf{M}(m^2n)\log(n)^2\big)$	$\mathcal{O}^{\sim}(n^{3\omega/2+3/2}k^{\omega/2+3/2})$

As said above, another approach rephrases the problem of computing Q in terms of polynomial matrix computations, that is, as linear algebra over $\mathbb{K}[X]$; this was in particular the basis of the extensions to the multivariate case $s > 1$ in [9, 8]. Starting from generators of an ad-hoc $\mathbb{K}[X]$-module (or polynomial lattice) that is known to contain a non-trivial Q, the algorithms in [27, 9, 4, 8, 5, 12] compute a Gröbner basis of that lattice, or simply a short vector therein. To achieve quasi-linear time in n, the algorithms in [4, 8] use a short vector subroutine due to Alekhnovich [1], while those in [5, 12] rely on a faster, randomized algorithm due to Giorgi, Jeannerod, and Villard [19].

Two main lattice constructions exist in the literature; following [9, §4.5], we present them directly in the case $s \geqslant 1$, and then give the cost bound that can be obtained using polynomial lattice reduction to find a short vector in the lattice. The first construction may be called *banded* (due to the shape of the generators it involves when $s = 1$); its generators derive from the polynomials G and $\boldsymbol{R}$ introduced before:

$$\left\{ G^i \prod_{r=1}^{s} (Y_r - R_r)^{j_r} \;\middle|\; i > 0,\ j_1, \ldots, j_s \geqslant 0,\ i + |\boldsymbol{j}| = m \right\}$$
$$\bigcup \quad \left\{ \prod_{r=1}^{s} (Y_r - R_r)^{j_r} Y_r^{J_r} \;\middle|\; j_1, \ldots, j_s \geqslant 0,\ J_1, \ldots, J_s \geqslant 0,\ |\boldsymbol{j}| = m,\ |\boldsymbol{J}| \leqslant \ell - m \right\}.$$

The second construction may be called *triangular*; its generators derive from the polynomials

$$\left\{ G^i \prod_{r=1}^{s} (Y_r - R_r)^{j_r} \;\middle|\; i > 0,\ j_1, \ldots, j_s \geqslant 0,\ i + |\boldsymbol{j}| = m \right\}$$
$$\bigcup \quad \left\{ \prod_{r=1}^{s} (Y_r - R_r)^{j_r} \;\middle|\; j_1, \ldots, j_s \geqslant 0,\ m \leqslant |\boldsymbol{j}| \leqslant \ell \right\}.$$

When $s = 1$, the first construction is used in [4, Remark 16] and [27, 12], and the second one is used in [4, 5]; the latter also appears in [8] for $s \geqslant 1$. In both cases the actual lattice bases are the coefficient vectors (in $\boldsymbol{Y}$) of the polynomials $h(X, X^k Y_1, \ldots, X^k Y_s)$, for h in either of the sets above; these X^k are introduced to account for the weighted-degree condition (iii).

In this context, for a lattice of dimension L given by generators of degree at most d, the algorithm in [19] computes a shortest vector in the lattice in expected time $\mathcal{O}(L^\omega \mathsf{M}(d) \log(Ld))$, as detailed below. For a deterministic solution, one can use the algorithm of Gupta, Sarkar, Storjohann, and Valeriote [20], whose cost is in $\mathcal{O}(L^\omega \mathsf{M}(d)((\log(L))^2 + \log(d)))$.)

For the banded basis, its dimension L_B and degree d_B can be taken as follows:

$$L_B = \binom{s+m-1}{s} + \binom{s+m-1}{s-1}\binom{s+\ell-m}{s} \quad \text{and} \quad d_B = \mathcal{O}(mn).$$

The dimension formula is given explicitly in [9, p. 75], while the degree bound is easily obtained when assuming that the parameters m, n, b of Problem 1 satisfy $b \leqslant mn$; such an assumption is not restrictive, since when $b > mn$ the polynomial $Q = G^m$ is a trivial solution. In this case, the arithmetic cost for constructing the lattice matrix with the given generators is $\mathcal{O}\left(\binom{s+m}{s}^2 \mathsf{M}(mn)\right)$, which is $\mathcal{O}(L_B^2\, \mathsf{M}(mn))$. Similarly, in the triangular case,

$$L_T = \binom{s+\ell}{s} \quad \text{and} \quad d_T = \mathcal{O}(\ell n),$$

and the cost for constructing the lattice matrix is $\mathcal{O}(L_T^2\, \mathsf{M}(\ell n))$.

Under our assumption $\mathbf{H_1} : m \leqslant \ell$, we always have $L_B \geqslant L_T$ and $d_B \leqslant d_T$; when $s = 1$, we get $L_B = L_T = \ell + 1$.

To bound the cost of reducing these two polynomial lattice bases, recall that the algorithm of [19] works as follows. Given a basis of a lattice of dimension L and degree d, if $x_0 \in \mathbb{K}$ is given such that the determinant of the lattice does not vanish at $X = x_0$, then the basis will be reduced deterministically using $\mathcal{O}(L^\omega \mathsf{M}(d) \log(Ld))$ operations in $\mathbb{K}$. Otherwise, such an x_0 is picked at random in $\mathbb{K}$ or, if the cardinality $|\mathbb{K}|$ is too small to ensure success with probability at least $1/2$, in a field extension $\mathbb{L}$ of $\mathbb{K}$. In general, $\mathbb{L}$ should be taken of degree $\mathcal{O}(\log(Ld))$ over $\mathbb{K}$; however, here degree 2 will suffice. Indeed, following [5, p. 206] we note that for the two lattice constructions above the determinants have the special form $G(X)^{i_1} X^{i_2}$ for some $i_1, i_2 \in \mathbb{N}$ (see Appendix C). Since $G(X) = (X - x_1) \cdots (X - x_n)$ with $x_1, \ldots, x_n \in \mathbb{K}$ pairwise distinct, x_0 can be found deterministically in time $\mathcal{O}(\mathsf{M}(n) \log(n))$

as soon as $|\mathbb{K}| > n+1$, by evaluating G at $n+1$ arbitrary elements of $\mathbb{K}$; else, $|\mathbb{K}|$ is either n or $n+1$, and x_0 can be found in an extension $\mathbb{L}$ of $\mathbb{K}$ of degree 2. Such an extension can be computed with probability of success at least $1/2$ in time $\mathcal{O}(\log(n))$ (see for example [18, §14.9]). Then, with the algorithm of [19] we obtain a reduced basis over $\mathbb{L}[X]$ using $\mathcal{O}(L^\omega \mathsf{M}(d)\log(Ld))$ operations in $\mathbb{L}$; since the degree of $\mathbb{L}$ over $\mathbb{K}$ is $\mathcal{O}(1)$, this is $\mathcal{O}(L^\omega \mathsf{M}(d)\log(Ld))$ operations in $\mathbb{K}$. Eventually, one can use [39, Theorems 13 and 20] to transform this basis into a reduced basis over $\mathbb{K}[X]$ without impacting the cost bound; or more directly, since here we are only looking for a sufficiently short vector in the lattice, this vector can be extracted from a shortest vector in the reduced basis over $\mathbb{L}[X]$. Therefore, by applying the algorithm of [19] to reduce the banded basis and triangular basis shown above, we will always obtain a polynomial Q solution to Problem 1 (assuming one exists) in expected time

$$\mathcal{O}(L_B^\omega \mathsf{M}(mn)\log(L_B mn)) \quad \text{and} \quad \mathcal{O}(L_T^\omega \mathsf{M}(\ell n)\log(L_T \ell n)),$$

respectively. For $s=1$ and under the assumption $\mathbf{H_1}$, these costs become $\mathcal{O}(\ell^\omega \mathsf{M}(mn)\log(\ell n))$ and $\mathcal{O}(\ell^\omega \mathsf{M}(\ell n)\log(\ell n))$, respectively, and are those reported in [12, 5]. For $s>1$, the costs reported in [9, 8] are worse, because the short vector algorithms used in those references are inferior to the ones we refer to. Under $\mathbf{H_1}$ and $\mathbf{H_2}$, the result in Theorem 2 is an improvement over those of both [9] and [8]. To see this, remark that the cost in our theorem is quasi-linear in $\binom{s+\ell}{s}^{\omega-1}\binom{s+m}{s+1}n$, whereas the costs in [9, 8] are at least $\binom{s+\ell}{s}^3 mn$; a simplification proves our claim.

It is interesting to notice that the two main approaches discussed here — solving a linear system over $\mathbb{K}$ or finding a short vector in a polynomial lattice — ultimately rely on the same condition on the parameters to ensure that a solution exists; this is detailed in Appendix C.

Regarding Problem 2, several particular cases of it are well known. When all P_i's are of the form $X^{M'_i}$, this problem becomes known as a simultaneous Hermite-Padé approximation problem or vector Hermite-Padé approximation problem [3, 42]. The case $\mu = 1$, with P_1 being given through its roots (and their multiplicities) is known as the M-Padé problem [2]. To our knowledge, the only previous work on Problem 2 in its full generality is by Nielsen in [31, Chapter 2]. Nielsen solves the problem by building an ad-hoc polynomial lattice, which has dimension $\mu+\nu$ and degree $\max_{i<\mu} M'_i$, and finding a short vector therein. Using the algorithm in [19], the overall cost bound for this approach is $\mathcal{O}^{\tilde{}}((\mu+\nu)^\omega(\max_{i<\mu} M'_i))$, to which our cost bound $\mathcal{O}^{\tilde{}}(\max(\mu,\nu)^{\omega-1}(\sum_{i<\mu} M'_i))$ from Theorem 1 compares favorably.

Outline of the paper. After a reminder on algorithms for structured linear systems in Section 2, we show how to reduce Problem 1 to Problem 2 in Section 3. This reduction is essentially based on Lemma 5, which extends to the multivariate case $s \geqslant 1$ the result in [47, Proposition 3]. Then, we give two algorithms that both prove Theorem 1, in Sections 4 and 5. The linearization in the first algorithm extends to the more general context of Problem 2 the derivation of Extended Key Equations presented in [47], ending up with a mosaic-Hankel system. The second algorithm gives a new approach for solving both problems, in which the linearization is more straightforward and the structure of the matrix of the system is

Toeplitz-like. Finally, Appendices A, B and C provide some insight regarding a few remarks made in Section 1.

2 Solving structured homogeneous linear systems

Our two solutions to Problem 2 rely on fast algorithms for solving linear systems of the form $Au = 0$ with A a structured matrix over $\mathbb{K}$. In this section, we briefly review useful concepts and results related to *displacement rank* techniques. While these techniques can handle systems with several kinds of structure, we will only need (and discuss) those related to *Toeplitz-like* and *Hankel-like* systems; for a more comprehensive treatment, the reader may consult [34].

Let M be a positive integer and let $\mathcal{Z}_M \in \mathbb{K}^{M\times M}$ be the square matrix with ones on the subdiagonal and zeros elsewhere:

$$\mathcal{Z}_M = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{K}^{M\times M}.$$

Given two integers M and N, consider the following operators:

$$\begin{array}{rccc} \Delta_{M,N}: & \mathbb{K}^{M\times N} & \to & \mathbb{K}^{M\times N} \\ & A & \mapsto & A - \mathcal{Z}_M\, A\, \mathcal{Z}_N^T \end{array}$$

and

$$\begin{array}{rccc} \Delta'_{M,N}: & \mathbb{K}^{M\times N} & \to & \mathbb{K}^{M\times N} \\ & A & \mapsto & A - \mathcal{Z}_M\, A\, \mathcal{Z}_N, \end{array}$$

which subtract from A its translate one place along the diagonal, resp. along the anti-diagonal.

Let us discuss $\Delta_{M,N}$ first. If A is a *Toeplitz* matrix, that is, invariant along diagonals, $\Delta_{M,N}(A)$ has rank at most two. As it turns out, Toeplitz systems can be solved much faster than general linear systems, in quasi-linear time in M. The main idea behind algorithms for structured matrices is to extend these algorithmic properties to those matrices A for which the rank of $\Delta_{M,N}(A)$ is small, in which case we say that A is *Toeplitz-like*. Below, this rank will be called the *displacement rank* of A (with respect to $\Delta_{M,N}$).

Two matrices (V, W) in $\mathbb{K}^{M\times\alpha} \times \mathbb{K}^{\alpha\times N}$ will be called a *generator of length* α for A with respect to $\Delta_{M,N}$ if $\Delta_{M,N}(A) = V\,W$. For the structure we are considering, one can recover A from its generators; in particular, one can use a generator of length α as a way to represent A using $\alpha(M + N)$ field elements. One of the main aspects of structured linear algebra algorithms is to use generators as a compact data structure throughout the whole process.

Up to now, we only discussed the Toeplitz structure. *Hankel-like* matrices are those which have a small displacement rank with respect to $\Delta'_{M,N}$, that is, those matrices A for which

the rank of $\Delta'_{M,N}(A)$ is small. As far as solving the system $Au = 0$ is concerned, this case can easily be reduced to the Toeplitz-like case. Define $B = AJ_N$, where J_N is the reversal matrix of size N, all entries of which are zero, except the anti-diagonal which is set to one. Then, one easily checks that the displacement rank of A with respect to $\Delta'_{M,N}$ is the same as the displacement rank of B with respect to $\Delta_{M,N}$, and that if (V, W) is a generator for A with respect to $\Delta'_{M,N}$, then (V, WJ_N) is a generator for B with respect to $\Delta_{M,N}$. Using the algorithm for Toeplitz-like matrices gives us a solution v to $Bv = 0$, from which we deduce that $u = J_N v$ is a solution to $Au = 0$.

In this paper, we will not enter the details of algorithms for solving such structured systems. The main result we will rely on is the following proposition, a minor extension of a result by Bostan, Jeannerod, and Schost [7], which features the best known complexity for this kind of task, to the best of our knowledge. This algorithm is based on previous work of Bitmead and Anderson [6], Morf [30], Kaltofen [24], and Pan [34], and is probabilistic (it depends on the choice of some parameters in the base field $\mathbb{K}$, and success is ensured provided these parameters avoid a hypersurface of the parameter space).

The proof of the following proposition occupies the rest of this section. Remark that some aspects of this statement could be improved (for instance, we could reduce the cost so that it only depends on M, not $\max(M, N)$), but that would be inconsequential for the applications we make of it.

Proposition 4. *Given a generator (V, W) of length α for a matrix $A \in \mathbb{K}^{M\times N}$, with respect to either $\Delta_{M,N}$ or $\Delta'_{M,N}$, one can find a nonzero element in the right nullspace of A, or determine that none exists, by a probabilistic algorithm that uses $\mathcal{O}(\alpha^{\omega-1}\mathsf{M}(P)\log(P)^2)$ operations in $\mathbb{K}$, with $P = \max(M, N)$. The algorithm chooses $\mathcal{O}(P)$ elements in $\mathbb{K}$; if these elements are chosen uniformly at random in a set $S \subseteq \mathbb{K}$ of cardinality at least $6P^2$, the probability of success is at least $1/2$.*

Square matrices. In all that follows, we consider only the operator $\Delta_{M,N}$, since we already pointed out that the case of $\Delta'_{M,N}$ can be reduced to it for no extra cost.

When $M = N$, we use directly [7, Theorem 1], which gives the running time reported above. That result does not explicitly state which solution we obtain, as it is written for general non-homogeneous systems. Here, we want to make sure we obtain a nonzero element in the right nullspace (if one exists), so slightly more details are needed.

The algorithm in that theorem chooses $3M - 2$ elements in $\mathbb{K}$, the first $2M - 2$ of which are used to precondition A by giving it generic rank profile; this is the case when these parameters avoid a hypersurface of $\mathbb{K}^{2M-2}$ of degree at most $M^2 + M$.

Assume this is the case. Then, following [25], the output vector u is obtained in a parametric form as $u = \lambda(u')$, where u' consists of another set of M parameters chosen in $\mathbb{K}$ and λ is a surjective linear mapping with image the right nullspace $\ker(A)$ of A. If $\ker(A)$ is trivial, the algorithm returns the zero vector in any case, which is correct. Otherwise, the set of vectors u' such that $\lambda(u') = 0$ is contained in a hyperplane of $\mathbb{K}^M$, so it is enough to choose u' outside of that hyperplane to ensure success.

To conclude we rely on the so-called Zippel-Schwartz lemma [14, 48, 40], which can be summarized as follows: if a nonzero polynomial over $\mathbb{K}$ of total degree at most d is evaluated by assigning each of its indeterminates a value chosen uniformly at random in a subset S of $\mathbb{K}$, then the probability that the resulting polynomial value be zero is at most $d/|S|$. Thus, applying that result to the polynomial of degree $d := M^2+M+1 \leqslant 3M^2$ corresponding to the hypersurface and the hyperplane mentioned above, we see that if we choose all parameters uniformly at random in a subset $S \subseteq \mathbb{K}$ of cardinality $|S| \geqslant 6M^2$, the algorithm succeeds with probability at least $1/2$.

Wide matrices. Suppose now that $M < N$, so that the system is underdetermined. We add $N - M$ zero rows on top of A, obtaining an $N \times N$ matrix A'. Applying the algorithm for the square case to A', we will obtain a right nullspace element u for A' and thus A, since these nullspaces are the same. In order to do so, we need to construct a generator for A' from the generator (V, W) we have for A: one simply takes (V', W), where V' is the matrix in $\mathbb{K}^{N\times\alpha}$ obtained by adding $N - M$ zero rows on top of V.

Tall matrices. Suppose finally that $M > N$. This time, we build the matrix $A' \in \mathbb{K}^{M\times M}$ by adjoining $M-N$ zero columns to A on the left. The generator (V, W) of A can be turned into a generator of A' by simply adjoining $M - N$ zero columns to W on the left. We then solve the system $A's = 0$, and return the vector u obtained by discarding the first $M - N$ entries of s.

The cost of this algorithm fits into the requested bound; all that remains to see is that we obtain a nonzero vector in the right nullspace $\ker(A)$ of A with nonzero probability. Indeed, the nullspaces of A and A' are now related by the equality $\ker(A') = \mathbb{K}^{M-N} \times \ker(A)$. We mentioned earlier that in the algorithm for the square case, the solution s to $A's = 0$ is obtained in parametric form, as $s = \lambda(s')$ for $s' \in \mathbb{K}^M$, with λ a surjective mapping $\mathbb{K}^M \to \ker(A')$. Composing with the projection $\pi : \ker(A') \to \ker(A)$, we obtain a parametrization of $\ker(A)$ as $u = (\pi \circ \lambda)(s')$. The error probability analysis is then the same as in the square case.

3 Reducing Problem 1 to Problem 2

In this section, we show how instances of Problem 1 can be reduced to instances of Problem 2; Algorithm 1 below gives an overview of this reduction. The main technical ingredient, stated in Lemma 5 below, generalizes to any $s \geqslant 1$ the one given for $s = 1$ by Zeh, Gentner, and Augot in [47, Proposition 3]. To prove it, we use the same steps as in [47]; we rely on the notion of Hasse derivatives, which allows us to write Taylor expansions in positive characteristic (see Hasse [23] or Roth [37, pp. 87, 276]).

In what follows, comparison and addition of multi-indices in $\mathbb{N}^s$ are defined componentwise. For example, writing $\boldsymbol{i} \leqslant \boldsymbol{j}$ is equivalent to ($i_1 \leqslant j_1$ and ... and $i_s \leqslant j_s$), and $\boldsymbol{i} - \boldsymbol{j}$ denotes $(i_1 - j_1, \ldots, i_s - j_s)$. If $\boldsymbol{y} = (y_1, \ldots, y_s)$ is in $\mathbb{K}[X]^s$ and $\boldsymbol{i} = (i_1, \ldots, i_s)$ is in $\mathbb{N}^s$,

then $\boldsymbol{Y} - \boldsymbol{y} = Y_1 - y_1, \ldots, Y_s - y_s$ and $\boldsymbol{Y}^{\boldsymbol{i}} = Y_1^{i_1} \cdots Y_s^{i_s}$. Finally, for products of binomial coefficients, we shall write

$$\binom{\boldsymbol{j}}{\boldsymbol{i}} = \binom{j_1}{i_1} \cdots \binom{j_s}{i_s}.$$

Note that this integer is zero when $\boldsymbol{i} \not\leqslant \boldsymbol{j}$.

If $\mathbb{A}$ is any commutative ring with unity and $\mathbb{A}[\boldsymbol{Y}]$ denotes the ring of polynomials in $Y_1, \ldots, Y_s$ over $\mathbb{A}$, then for a polynomial $P(\boldsymbol{Y}) = \sum_{\boldsymbol{j}} P_{\boldsymbol{j}} \boldsymbol{Y}^{\boldsymbol{j}}$ in $\mathbb{A}[\boldsymbol{Y}]$ and a multi-index $\boldsymbol{i}$ in $\mathbb{N}^s$, the *order-$\boldsymbol{i}$ Hasse derivative* of P is the polynomial $P^{[\boldsymbol{i}]}$ in $\mathbb{A}[\boldsymbol{Y}]$ defined by

$$P^{[\boldsymbol{i}]} = \sum_{\boldsymbol{j} \geqslant \boldsymbol{i}} \binom{\boldsymbol{j}}{\boldsymbol{i}} P_{\boldsymbol{j}} \boldsymbol{Y}^{\boldsymbol{j}-\boldsymbol{i}}.$$

The Hasse derivative satisfies the following property (Taylor expansion): for all $\boldsymbol{a}$ in $\mathbb{A}^s$,

$$P(\boldsymbol{Y}) = \sum_{\boldsymbol{i}} P^{[\boldsymbol{i}]}(\boldsymbol{a})(\boldsymbol{Y} - \boldsymbol{a})^{\boldsymbol{i}}.$$

The next lemma shows how Hasse derivatives can help rephrase the vanishing condition (iv) of Problem 1. Below, $G \in \mathbb{K}[X]$ and $\boldsymbol{R} \in \mathbb{K}[X]^s$ are the master polynomial and the Lagrange polynomials which were defined in (1) and (2) in the introduction.

Lemma 5. *For any polynomial Q in $\mathbb{K}[X, \boldsymbol{Y}]$, Q satisfies the condition* (iv) *of Problem 1 if and only if for all $\boldsymbol{i}$ in $\mathbb{N}^s$ such that $|\boldsymbol{i}| < m$,*

$$Q^{[\boldsymbol{i}]}(X, \boldsymbol{R}) = 0 \bmod G^{m-|\boldsymbol{i}|}.$$

Proof. Since the x_r's defining $G = \prod_{r=1}^{n}(X - x_r)$ are pairwise distinct, it suffices to prove, for $1 \leqslant r \leqslant n$, the following equivalence for the point $(x_r, \boldsymbol{y}_r)$: $Q(x_r, \boldsymbol{y}_r) = 0$ with multiplicity at least m if and only if for all $\boldsymbol{i}$ in $\mathbb{N}^s$ such that $|\boldsymbol{i}| < m$, $Q^{[\boldsymbol{i}]}(X, \boldsymbol{R}) = 0 \bmod (X - x_r)^{m-|\boldsymbol{i}|}$. Now, up to a shift one can assume that this point is $\boldsymbol{0} \in \mathbb{K}^{s+1}$; in other words, it suffices to show that for $\boldsymbol{R}(0) = \boldsymbol{0} \in \mathbb{K}^s$, we have $Q(0, \boldsymbol{0}) = 0$ with multiplicity at least m if and only if, for all $\boldsymbol{i}$ in $\mathbb{N}^s$ such that $|\boldsymbol{i}| < m$, $X^{m-|\boldsymbol{i}|}$ divides $Q^{[\boldsymbol{i}]}(X, \boldsymbol{R})$.

Assume first that $\boldsymbol{0} \in \mathbb{K}^{s+1}$ is a root of Q of multiplicity at least m. Then, $Q(X, \boldsymbol{Y}) = \sum_{\boldsymbol{j}} Q_{\boldsymbol{j}} \boldsymbol{Y}^{\boldsymbol{j}}$ has only monomials of total degree at least m, so that for $\boldsymbol{j} \geqslant \boldsymbol{i}$, each nonzero $Q_{\boldsymbol{j}} \boldsymbol{Y}^{\boldsymbol{j}-\boldsymbol{i}}$ has only monomials of total degree at least $m - |\boldsymbol{i}|$. Now, $\boldsymbol{R}(0) = \boldsymbol{0} \in \mathbb{K}^s$ implies that X divides each component of $\boldsymbol{R}$. Consequently, $X^{m-|\boldsymbol{i}|}$ divides $Q_{\boldsymbol{j}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}}$ for each $\boldsymbol{j} \geqslant \boldsymbol{i}$, and thus $Q^{[\boldsymbol{i}]}(X, \boldsymbol{R})$ as well.

Conversely, let us assume that for all $\boldsymbol{i}$ in $\mathbb{N}^s$ such that $|\boldsymbol{i}| < m$, $X^{m-|\boldsymbol{i}|}$ divides $Q^{[\boldsymbol{i}]}(X, \boldsymbol{R})$, and show that Q has no monomial of total degree less than m. Writing the Taylor expansion of Q with $\mathbb{A} = \mathbb{K}[X]$ and $\boldsymbol{a} = \boldsymbol{R}$, we obtain

$$Q(X, \boldsymbol{Y}) = \sum_{\boldsymbol{i}} Q^{[\boldsymbol{i}]}(X, \boldsymbol{R})(\boldsymbol{Y} - \boldsymbol{R})^{\boldsymbol{i}}.$$

Each component of $\boldsymbol{R}$ being a multiple of X, we deduce that for the multi-indices $\boldsymbol{i}$ such that $|\boldsymbol{i}| \geqslant m$ every nonzero monomial in $Q^{[\boldsymbol{i}]}(X, \boldsymbol{R})(\boldsymbol{Y} - \boldsymbol{R})^{\boldsymbol{i}}$ has total degree at least m. Using our assumption, the same conclusion follows for the multi-indices such that $|\boldsymbol{i}| < m$. □

Writing $\boldsymbol{k} \cdot \boldsymbol{j} = k_1 j_1 + \cdots + k_s j_s$, recall from the statement of Theorem 2 that Γ is the set of all $\boldsymbol{j}$ in $\mathbb{N}^s$ such that $|\boldsymbol{j}| \leqslant \ell$ and $\boldsymbol{k} \cdot \boldsymbol{j} < b$. Then, defining the positive integers

$$N_{\boldsymbol{j}} = b - \boldsymbol{k} \cdot \boldsymbol{j}$$

for all $\boldsymbol{j}$ in Γ, we immediately obtain the following reformulation of the list-size and weighted-degree conditions of our interpolation problem:

Lemma 6. *For any polynomial Q in $\mathbb{K}[X, \boldsymbol{Y}]$, Q satisfies the conditions* (ii) *and* (iii) *of Problem 1 if and only if it has the form*

$$Q(X, \boldsymbol{Y}) = \sum_{\boldsymbol{j} \in \Gamma} Q_{\boldsymbol{j}}(X) \boldsymbol{Y}^{\boldsymbol{j}} \quad \textit{with} \quad \deg(Q_{\boldsymbol{j}}) < N_{\boldsymbol{j}}.$$

For $\boldsymbol{i} \in \mathbb{N}^s$ with $|\boldsymbol{i}| < m$ and $\boldsymbol{j} \in \Gamma$, let us now define the polynomials $P_{\boldsymbol{i}}, F_{\boldsymbol{i},\boldsymbol{j}} \in \mathbb{K}[X]$ as

$$P_{\boldsymbol{i}} = G^{m-|\boldsymbol{i}|} \qquad \text{and} \qquad F_{\boldsymbol{i},\boldsymbol{j}} = \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} \bmod P_{\boldsymbol{i}}. \tag{3}$$

It then follows from Lemmas 5 and 6 that Q in $\mathbb{K}[X, \boldsymbol{Y}]$ satisfies the conditions (ii), (iii), (iv) of Problem 1 if and only if $Q = \sum_{\boldsymbol{j} \in \Gamma} Q_{\boldsymbol{j}} \boldsymbol{Y}^{\boldsymbol{j}}$ for some polynomials $Q_{\boldsymbol{j}}$ in $\mathbb{K}[X]$ such that

- $\deg(Q_{\boldsymbol{j}}) < N_{\boldsymbol{j}}$ for all $\boldsymbol{j}$ in Γ,
- $\sum_{\boldsymbol{j} \in \Gamma} Q_{\boldsymbol{j}} F_{\boldsymbol{i},\boldsymbol{j}} = 0 \bmod P_{\boldsymbol{i}}$ for all $|\boldsymbol{i}| < m$.

Since the $P_{\boldsymbol{i}}$ are monic polynomials of degree $M_{\boldsymbol{i}} := n(m - |\boldsymbol{i}|)$ and since $\deg F_{\boldsymbol{i},\boldsymbol{j}} < M_{\boldsymbol{i}}$, the latter conditions express the problem of finding such a Q as an instance of Problem 2. In order to make the reduction completely explicit, define further

$$M = \sum_{|\boldsymbol{i}| < m} M_{\boldsymbol{i}},$$

$$\mu = \binom{s+m-1}{s}, \qquad \nu = |\Gamma|, \qquad \varrho = \max(\mu, \nu),$$

and choose arbitrary orders on the sets of indices $\{\boldsymbol{i} \in \mathbb{N}^s \mid |\boldsymbol{i}| < m\}$ and Γ, that is, bijections

$$\phi : \{0, \ldots, \mu - 1\} \to \{\boldsymbol{i} \in \mathbb{N}^s \mid |\boldsymbol{i}| < m\} \quad \text{and} \quad \psi : \{0, \ldots, \nu - 1\} \to \Gamma. \tag{4}$$

Then, for i in $\{0, \ldots, \mu - 1\}$ and j in $\{0, \ldots, \nu - 1\}$, we associate $M'_i = M_{\phi(i)}$, $N'_j = N_{\psi(j)}$, $P'_i = P_{\phi(i)}$ and $F'_{i,j} = F_{\phi(i),\psi(j)}$.

Proposition 7. *Given positive integers $s, \ell, m, n, b, k_1, \ldots, k_s$, let $\mu, \nu, M'_i, N'_j, P'_i, F'_{i,j}$ as well as ϱ and M be defined as above. Then, using $\mathcal{O}(\varrho\, \mathsf{M}(M) \log(M))$ operations in $\mathbb{K}$, one can reduce an instance of Problem 1 with parameters s, ℓ, m, n, b, $k_1, \ldots, k_s$ and points $\{(x_i, y_{i,1}, \ldots, y_{i,s})\}_{1 \leqslant i \leqslant n}$ to an instance of Problem 2 with parameters $\mu, \nu, M'_0, \ldots, M'_{\mu-1}$, $N'_0, \ldots, N'_{\nu-1}$ and input polynomials $\{(P'_i, F'_{i,0}, \ldots, F'_{i,\nu-1})\}_{0 \leqslant i < \mu}$.*

Proof. The only thing left to do is the complexity analysis. First, we need to compute $P_{\boldsymbol{i}} = G^{m-|\boldsymbol{i}|}$ for every $\boldsymbol{i}$ such that $|\boldsymbol{i}| < m$. This involves only m different polynomials, namely $G, G^2, \ldots, G^m$, so it can be done using $\mathcal{O}(m\mathsf{M}(mn))$ operations; this cost is in $\mathcal{O}(\varrho\,\mathsf{M}(M))$, since $\binom{s+m-1}{s} \geqslant m$ and $M = \sum_{|\boldsymbol{i}|<m} n(m-|\boldsymbol{i}|) \geqslant mn$.

Then, we have to compute (some of) the interpolation polynomials $R_1, \ldots, R_s$. Due to Lemma 5, the only values of $i \in \{1, \ldots, s\}$ for which R_i is needed are those such that the indeterminate Y_i may actually appear in $Q(X, \boldsymbol{Y}) = \sum_{\boldsymbol{j}\in\Gamma} Q_{\boldsymbol{j}}(X)\boldsymbol{Y}^{\boldsymbol{j}}$. Now, the latter will not occur unless the ith unit s-tuple $(0, \ldots, 0, 1, 0, \ldots, 0)$ belongs to Γ. Hence, at most $|\Gamma|$ polynomials R_i must be computed, each at a cost of $\mathcal{O}(\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$. Overall, the cost of the interpolation step is thus in $\mathcal{O}(|\Gamma|\mathsf{M}(n)\log(n)) \subset \mathcal{O}(\varrho\,\mathsf{M}(M)\log(M))$.

Finally, we compute $F_{\boldsymbol{i},\boldsymbol{j}}$ for every $\boldsymbol{i}, \boldsymbol{j}$. This is done by fixing $\boldsymbol{i}$ and computing all products $F_{\boldsymbol{i},\boldsymbol{j}}$ incrementally, starting from $R_1, \ldots, R_s$. Since we compute modulo $P_{\boldsymbol{i}}$, each product takes $\mathcal{O}(\mathsf{M}(M_{\boldsymbol{i}}))$ operations in $\mathbb{K}$. Summing over all $\boldsymbol{j}$ leads to a cost of $\mathcal{O}(|\Gamma|\mathsf{M}(M_{\boldsymbol{i}}))$ per index $\boldsymbol{i}$. Summing over all $\boldsymbol{i}$ and using the super-linearity of M leads to a total cost of $\mathcal{O}(|\Gamma|\mathsf{M}(M))$, which is $\mathcal{O}(\varrho\mathsf{M}(M))$. □

The reduction above is deterministic and its cost is negligible compared to the cost in $\mathcal{O}(\varrho^{\omega-1}\,\mathsf{M}(M)\log(M)^2)$ that follows from Theorem 1 with $\rho = \varrho$ and $M' = \sum_{0\leqslant i<\mu} M'_i = M$. Since M is by definition equal to $\binom{s+m}{s+1}n$, we conclude that Theorem 1 implies Theorem 2.

Algorithm 1. Reduction from Problem 1 to Problem 2.

Input: s, ℓ, m, n, b in $\mathbb{N}_{>0}$, $(k_1, \ldots, k_s)$ in $\mathbb{N}_{>0}^s$ and points $\{(x_i, y_{i,1}, \ldots, y_{i,s})\}_{1\leqslant i\leqslant n}$ in $\mathbb{K}^{s+1}$ with the x_i's pairwise distinct.

Output: parameters $\mu,\ \nu,\ M'_0, \ldots, M'_{\mu-1},\ N'_0, \ldots, N'_{\nu-1},\ \{(P_i, F_{i,0}, \ldots, F_{i,\nu-1})\}_{0\leqslant i<\mu}$ for Problem 2, such that a solution to this problem is a solution to Problem 1 with parameters the input of this algorithm.

1. Compute $G(X)$ and $R_1(X), \ldots, R_s(X)$ as in (1) and (2)

2. Compute $M_{\boldsymbol{i}} = n(m - |\boldsymbol{i}|)$ for $|\boldsymbol{i}| < m$, and $\mu = \binom{s+m-1}{s}$

3. Compute $N_{\boldsymbol{j}} = b - \boldsymbol{k}\cdot\boldsymbol{j}$ for $\boldsymbol{j} \in \Gamma = \{\boldsymbol{j} \in \mathbb{N}^s \mid |\boldsymbol{j}| \leqslant \ell \text{ and } N_{\boldsymbol{j}} > 0\}$, and $\nu = |\Gamma|$

4. Compute $P_{\boldsymbol{i}}$ and $F_{\boldsymbol{i},\boldsymbol{j}}$ for $|\boldsymbol{i}| < m$, $\boldsymbol{j} \in \Gamma$ as in (3)

5. Compute bijections ϕ and ψ as in (4)

6. Return the integers $\mu, \nu, M_{\phi(0)}, \ldots, M_{\phi(\mu-1)}, N_{\psi(0)}, \ldots, N_{\psi(\nu-1)}$, and the polynomials $\{(P_{\phi(i)}, F_{\phi(i),\psi(0)}, \ldots, F_{\phi(i),\psi(\nu-1)})\}_{0\leqslant i<\mu}$

4 Solving Problem 2 through a mosaic-Hankel linear system

In this section, we give our first solution to Problem 2, thereby proving Theorem 1; this solution is outlined in Algorithm 2 below. It consists of deriving and then linearizing the modular equations of Lemma 8 below, which can be seen as a generalization of the so-called key equations in the context of Reed-Solomon codes. In particular when solving Problem 1 using the reduction to Problem 2 exposed in Section 3, the equations of Lemma 8 extend to arbitrary s the extended key equations presented in [38, 47] for $s = 1$. However, unlike in those references, we apply the approach recalled in Section 2 to solve the resulting mosaic-Hankel linear system, since it features the best cost we are aware of for this task.

We consider input polynomials $\{(P_i, \boldsymbol{F}_i)\}_{0 \leqslant i < \mu}$ with, for all i, P_i monic of degree M'_i and $\boldsymbol{F}_i$ a vector of ν polynomials $(F_{i,0}, \ldots, F_{i,\nu-1})$, all of degree less than M'_i. Given degree bounds $N'_0, \ldots, N'_{\nu-1}$, we look for polynomials $\boldsymbol{Q} = (Q_0, \ldots, Q_{\nu-1})$ in $\mathbb{K}[X]$ such that the following holds:

(a) the Q_j's are not all zero,

(b) for $0 \leqslant j < \nu$, $\deg(Q_j) < N'_j$,

(c) for $0 \leqslant i < \mu$, $\sum_{0 \leqslant j < \nu} Q_j F_{i,j} = 0 \bmod P_i$.

Our goal here is to linearize Problem 2 into a homogeneous linear system over $\mathbb{K}$ involving M' linear equations with N' unknowns, where $M' = M'_0 + \cdots + M'_{\mu-1}$ and $N' = N'_0 + \cdots + N'_{\nu-1}$. Without loss of generality, we will assume that

$$N' \leqslant M' + 1.$$

Indeed, if $N' \geqslant M'+1$, the instance of Problem 2 we are considering has more unknowns than equations. We may set the last $N' - (M' + 1)$ unknowns to zero, while keeping the system underdetermined. This simply amounts to replacing the degree bounds $N'_0, \ldots, N'_{\nu-1}$ by $N'_0, \ldots, N'_{\nu'-2}, N''_{\nu'-1}$, for $\nu' \leqslant \nu$ and $N''_{\nu'-1} \leqslant N'_{\nu'-1}$ such that $N'_0 + \cdots + N'_{\nu'-2} + N''_{\nu'-1} = M'+1$. In particular, ν may only decrease through this process.

In what follows, we will work with the reversals of the input and output polynomials of Problem 2, defined by

$$\overline{P_i} = X^{M'_i} P_i(X^{-1}), \qquad \overline{F_{i,j}} = X^{M'_i - 1} F_{i,j}(X^{-1}), \qquad \overline{Q_j} = X^{N'_j - 1} Q_j(X^{-1}).$$

Let also $\beta = \max_{h<\nu} N'_h$ and, for $0 \leqslant i < \mu$ and $0 \leqslant j < \nu$,

$$\delta_i = M'_i + \beta - 1 \quad \text{and} \quad \gamma_j = \beta - N'_j.$$

In particular, δ_i and γ_j are nonnegative integers and, recalling that P_i is monic, we can define further the polynomials $S_{i,j}$ in $\mathbb{K}[X]$ as

$$S_{i,j} = \frac{X^{\gamma_j} \overline{F_{i,j}}}{\overline{P_i}} \bmod X^{\delta_i}$$

for $0 \leqslant i < \mu$ and $0 \leqslant j < \nu$. By using these polynomials, we can now reformulate the approximation condition of Problem 2 in terms of a set of extended key equations:

Lemma 8. *Let $Q_0, \ldots, Q_{\nu-1}$ be polynomials in $\mathbb{K}[X]$ that satisfy condition* (b) *in Problem 2. They satisfy condition* (c) *in Problem 2 if and only if for all i in $\{0, \ldots, \mu-1\}$, there exists a polynomial T_i in $\mathbb{K}[X]$ such that*

$$\sum_{0 \leqslant j < \nu} \overline{Q_j} S_{i,j} = T_i \bmod X^{\delta_i} \qquad \text{and} \qquad \deg(T_i) < \beta - 1. \tag{5}$$

Proof. Condition (c) holds if and only if for all i in $\{0, \ldots, \mu-1\}$, there exists a polynomial B_i in $\mathbb{K}[X]$ such that

$$\sum_{0 \leqslant j < \nu} Q_j F_{i,j} = B_i P_i. \tag{6}$$

For all i, j, the summand $Q_j F_{i,j}$ has degree less than $N'_j + M'_i - 1$, so the left-hand term above has degree less than δ_i. Since P_i has degree M'_i, this implies that whenever a polynomial B_i as above exists, we must have $\deg(B_i) < \delta_i - M'_i = \beta - 1$. Now, by substituting $1/X$ for X and multiplying by $X^{\delta_i - 1}$ we can rewrite the identity in (6) as

$$\sum_{0 \leqslant j < \nu} \overline{Q_j}\,\overline{F_{i,j}} X^{\gamma_j} = T_i \overline{P_i}, \tag{7}$$

where T_i is the polynomial of degree less than $\beta - 1$ given by $T_i = X^{\beta-2} B_i(X^{-1})$. Since the degrees of both sides of (7) are less than δ_i, one can consider the above identity modulo X^{δ_i} without loss of generality, and since $\overline{P_i}(0) = 1$ one can further divide by $\overline{P_i}$ modulo X^{δ_i}. This shows that (7) is equivalent to the identity in (5), and the proof is complete. □

Following [38, 47], we are going to rewrite the latter conditions as a linear system in the coefficients of the polynomials $Q_0, \ldots, Q_{\nu-1}$, eliminating the unknowns T_i from the outset. Let us first define the *coefficient vector* of a solution $(Q_0, \ldots, Q_{\nu-1})$ to Problem 2 as the vector in $\mathbb{K}^{N'}$ obtained by concatenating, for $0 \leqslant j < \nu$, the vectors $\left[Q_j^{(0)}, Q_j^{(1)}, \ldots, Q_j^{(N'_j-1)}\right]^T$ of the coefficients of Q_j. Furthermore, denoting by $S_{i,j}^{(0)}, S_{i,j}^{(1)}, \ldots, S_{i,j}^{(\delta_i-1)}$ the $\delta_i \geqslant 1$ coefficients of the polynomial $S_{i,j}$, we set up the block matrix

$$A = \left[A_{i,j}\right]_{0 \leqslant i < \mu,\, 0 \leqslant j < \nu} \in \mathbb{K}^{M' \times N'},$$

whose block (i, j) is the Hankel matrix

$$A_{i,j} = \left[S_{i,j}^{(u+v+\gamma_j)}\right]_{0 \leqslant u < M'_i,\, 0 \leqslant v < N'_j} \in \mathbb{K}^{M'_i \times N'_j}.$$

Lemma 9. *A nonzero vector of $\mathbb{K}^{N'}$ is in the right nullspace of A if and only if it is the coefficient vector of a solution $(Q_0, \ldots, Q_{\nu-1})$ to Problem 2.*

Proof. It is sufficient to consider a polynomial tuple $(Q_0, \ldots, Q_{\nu-1})$ that satisfies (b). Then, looking at the high-degree terms in the identities in (5), we see that condition (c) is equivalent to the following homogeneous system of linear equations over $\mathbb{K}$: for all i in $\{0, \ldots, \mu-1\}$ and all δ in $\{\delta_i - M'_i, \ldots, \delta_i - 1\}$,

$$\sum_{0 \leqslant j < \nu} \sum_{0 \leqslant r < N'_j} Q_j^{(N'_j - 1 - r)} S_{i,j}^{(\delta - r)} = 0.$$

The matrix obtained by considering all these equations is precisely the matrix A. □

We will use the approach recalled in Section 2 to find a nonzero nullspace element for A, with respect to the displacement operator $\Delta'_{M',N'}$. Not only do we need to prove that the displacement rank of A with respect to $\Delta'_{M',N'}$ is bounded by a value α not too large, but we also have to efficiently compute a generator of length α for A, that is, a matrix pair (V, W) in $\mathbb{K}^{M' \times \alpha} \times \mathbb{K}^{\alpha \times N'}$ such that $A - \mathcal{Z}_{M'} A \, \mathcal{Z}_{N'} = VW$. We will see that here, computing such a generator boils down to computing the coefficients of the polynomials $S_{i,j}$. The cost incurred by computing this generator is summarized in the following lemma; combined with Proposition 4 and Lemma 9, this proves Theorem 1.

Lemma 10. *The displacement rank of A with respect to $\Delta'_{M',N'}$ is at most $\mu + \nu$. Furthermore, a corresponding generator of length $\mu+\nu$ for A can be computed using $\mathcal{O}\left((\mu+\nu)\mathsf{M}(M')\right)$ operations in $\mathbb{K}$.*

Proof. We are going to exhibit two matrices $V \in \mathbb{K}^{M' \times (\mu+\nu)}$ and $W \in \mathbb{K}^{(\mu+\nu) \times N'}$ such that $A - \mathcal{Z}_{M'} A \, \mathcal{Z}_{N'} = VW$. Because of the structure of A, at most μ rows and ν columns of the matrix $A - \mathcal{Z}_{M'} A \, \mathcal{Z}_{N'} = A - (A$ shifted left and down by one unit) are nonzero. More precisely, only the first row and the last column of each $M'_i \times N'_j$ block of this matrix can be nonzero. Indexing the rows, resp. columns, of $A - \mathcal{Z}_{M'} A \, \mathcal{Z}_{N'}$ from 0 to $M'-1$, resp. from 0 to $N'-1$, only the μ rows with indices of the form $r_i = M'_0 + \cdots + M'_{i-1}$ for $i = 0, \ldots, \mu-1$ can be nonzero, and only the ν columns with indices of the form $c_j = N'_0 + \cdots + N'_j - 1$ for $j = 0, \ldots, \nu-1$ can be nonzero.

For any integers $0 \leqslant i < K$, define $\mathcal{O}_{i,K} = [0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0]^T \in \mathbb{K}^K$ with 1 at position i, and

$$\mathcal{O}^{(V)} = [\mathcal{O}_{r_i,M'}]_{0 \leqslant i < \mu} \in \mathbb{K}^{M' \times \mu}, \quad \mathcal{O}^{(W)} = \left[\mathcal{O}_{c_j,N'}\right]_{0 \leqslant j < \nu}^T \in \mathbb{K}^{\nu \times N'}.$$

For given i in $\{0, \ldots, \mu-1\}$ and j in $\{0, \ldots, \nu-1\}$, we will consider $v_{i,j} = [v_{i,j}^{(r)}]_{0 \leqslant r < M'_i}$ in $\mathbb{K}^{M'_i \times 1}$ and $w_{i,j} = [w_{i,j}^{(r)}]_{0 \leqslant r < N'_j}$ in $\mathbb{K}^{1 \times N'_j}$, which are respectively the last column and the first row of the block (i,j) in $A - \mathcal{Z}_{M'} A \, \mathcal{Z}_{N'}$, up to a minor point: the first entry of $v_{i,j}$ is set to zero. The coefficients $v_{i,j}^{(r)}$ and $w_{i,j}^{(r)}$ can then be expressed in terms of the entries

$A_{i,j}^{(u,v)} = S_{i,j}^{(u+v+\gamma_j)}$ of the Hankel matrix $A_{i,j} = [A_{i,j}^{(u,v)}]_{0 \leqslant u < M'_i, 0 \leqslant v < N'_j}$ as follows:

$$v_{i,j}^{(r)} = \begin{cases} 0 & \text{if } r = 0, \\ A_{i,j}^{(r,N'_j-1)} - A_{(i,j+1)}^{(r-1,0)} & \text{if } 1 \leqslant r < M'_i, \end{cases} \tag{8}$$

$$w_{i,j}^{(r)} = \begin{cases} A_{i,j}^{(0,r)} - A_{i-1,j}^{(M'_{i-1}-1,r+1)} & \text{if } r < N'_j - 1, \\ A_{i,j}^{(0,N'_j-1)} - A_{i-1,j+1}^{(M'_{i-1}-1,0)} & \text{if } r = N'_j - 1. \end{cases} \tag{9}$$

Note that here, we use the convention that an indexed object is zero when the index is out of the allowed bounds for this object.

Then, we define V_j and W_i as

$$V_j = \begin{bmatrix} v_{0,j} \\ \vdots \\ v_{\mu-1,j} \end{bmatrix} \in \mathbb{K}^{M' \times 1} \quad \text{and} \quad W_i = [w_{i,0} \ \cdots \ w_{i,\nu-1}] \in \mathbb{K}^{1 \times N'},$$

and

$$V' = [V_0 \cdots V_{\nu-1}] \in \mathbb{K}^{M' \times \nu} \quad \text{and} \quad W' = \begin{bmatrix} W_0 \\ \vdots \\ W_{\mu-1} \end{bmatrix} \in \mathbb{K}^{\mu \times N'}.$$

Now, one can easily verify that the matrices

$$V = \begin{bmatrix} V' & \mathcal{O}^{(V)} \end{bmatrix} \in \mathbb{K}^{M' \times (\mu+\nu)} \qquad \text{and} \qquad W = \begin{bmatrix} \mathcal{O}^{(W)} \\ W' \end{bmatrix} \in \mathbb{K}^{(\mu+\nu) \times N'} \tag{10}$$

are generators for A, that is, $A - \mathcal{Z}_M A \, \mathcal{Z}_N = VW$.

We notice that all we need to compute the generators V and W are the last $M'_i + N'_j - 1$ coefficients of $S_{i,j}(X) = S_{i,j}^{(0)} + S_{i,j}^{(1)} X + \cdots + S_{i,j}^{(\delta_i-1)} X^{\delta_i - 1}$ for every i in $\{0, \ldots, \mu - 1\}$ and j in $\{0, \ldots, \nu - 1\}$. Now, recall that

$$S_{i,j} = \frac{X^{\gamma_j} \overline{F_{i,j}}}{\overline{P_i}} \bmod X^{\delta_i} = \frac{X^{\delta_i - (M'_i + N'_j - 1)} \overline{F_{i,j}}}{\overline{P_i}} \bmod X^{\delta_i}.$$

Thus, the first $\delta_i - (M'_i + N'_j - 1)$ coefficients of $S_{i,j}$ are zero, and the last $M'_i + N'_j - 1$ coefficients of $S_{i,j}$ are the coefficients of

$$S^{\star}_{i,j} = \frac{\overline{F_{i,j}}}{\overline{P_i}} \bmod X^{M'_i + N'_j - 1}, \tag{11}$$

which can be computed in $\mathcal{O}(\mathsf{M}(M'_i + N'_j))$ operations in $\mathbb{K}$ by fast power series division. By expanding products, we see that $\mathsf{M}(M'_i + N'_j) = \mathcal{O}(\mathsf{M}(M'_i) + \mathsf{M}(N'_j))$. Summing the costs, we obtain an upper bound of the form

$$\mathcal{O}\left(\sum_{0 \leqslant i < \mu} \sum_{0 \leqslant j < \nu} \mathsf{M}(M'_i) + \mathsf{M}(N'_j) \right).$$

Using the super-linearity of M, this is in $\mathcal{O}(\nu\mathsf{M}(M') + \mu\mathsf{M}(N'))$. Since we assumed that $N' \leqslant M' + 1$, this is $\mathcal{O}((\mu+\nu)\mathsf{M}(M'))$. □

Algorithm 2. Solving Problem 2 via generalized Key Equations.

Input: positive integers $\mu,\ \nu,\ M'_0, \ldots, M'_{\mu-1},\ N'_0, \ldots, N'_{\nu-1}$ and polynomial tuples $\{(P_i, F_{i,0}, \ldots, F_{i,\nu-1})\}_{0\leqslant i<\mu}$ in $\mathbb{K}[X]^{\nu+1}$ such that for all i, P_i is monic of degree M'_i and $\deg(F_{i,j}) < M'_i$ for all j.

Output: polynomials $Q_0, \ldots, Q_{\nu-1}$ in $\mathbb{K}[X]$ such that (a), (b), (c).

1. For $i < \mu$ and $j < \nu$, compute the coefficients $\{S_{i,j}^{(\gamma_j+r)},\ r < M'_i + N'_j\}$, that is, the coefficients of $S^\star_{i,j}$ as defined in (11)

2. For $i < \mu$ and $j < \nu$, compute the vectors $v_{i,j}$ and $w_{i,j}$ as defined in (8) and (9)

3. For $i < \mu$, compute $r_i = M'_0 + \cdots + M'_{i-1}$; for $j < \nu$, compute $c_j = N'_0 + \cdots + N'_j - 1$

4. Deduce the generators V and W as defined in (10) from $r_i, c_j, v_{i,j}, w_{i,j}$

5. Use the algorithm of Proposition 4 with input V and W; if there is no solution then exit with no solution, otherwise find the coefficients of $Q_0, \ldots, Q_{\nu-1}$

6. Return $Q_0, \ldots, Q_{\nu-1}$

5 A direct solution to Problem 2

In this section, we propose an alternative solution to Problem 2 which leads to the same asymptotic running time as in the previous section but avoids the extended key equations of Lemma 8; it is outlined in Algorithm 3 below. As above, our input consists of the polynomials $(P_i, F_{i,0}, \ldots, F_{i,\nu-1})_{0\leqslant i<\mu}$ and we look for polynomials $Q_0, \ldots, Q_{\nu-1}$ in $\mathbb{K}[X]$ such that for $0 \leqslant i < \mu$, $\sum_{0\leqslant j<\nu} Q_j F_{i,j} = 0 \bmod P_i$, with the Q_j's not all zero and for $j < \nu$, $\deg Q_j < N'_j$.

In addition, for $r \geqslant 0$, we denote by $F_{i,j}^{(r)}$ and $P_i^{(r)}$ the coefficients of degree r of $F_{i,j}$ and P_i, respectively, and we define $\mathcal{C}_i$ as the $M'_i \times M'_i$ companion matrix of P_i; if B is a polynomial of degree less than M'_i with coefficient vector $v \in \mathbb{K}^{M'_i}$, then the product $\mathcal{C}_i v \in \mathbb{K}^{M'_i}$ is the coefficient vector of the polynomial $XB \bmod P_i$. Explicitly, we have

$$\mathcal{C}_i = \begin{bmatrix} 0 & 0 & \cdots & 0 & -P_i^{(0)} \\ 1 & 0 & \cdots & 0 & -P_i^{(1)} \\ 0 & 1 & \cdots & 0 & -P_i^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -P_i^{(M'_i-1)} \end{bmatrix} \in \mathbb{K}^{M'_i \times M'_i}.$$

We are going to see that solving Problem 2 is equivalent to finding a nonzero solution to a homogeneous linear system whose matrix is $A' = (A'_{i,j}) \in \mathbb{K}^{M'\times N'}$, where for every

$i < \mu$ and $j < \nu$, $A'_{i,j} \in \mathbb{K}^{M'_i \times N'_j}$ is a matrix which depends on the coefficients of $F_{i,j}$ and P_i. Without loss of generality, we make the same assumption as in the previous section, that is, $N' \leqslant M' + 1$ holds.

For i, j as above and for $h \in \mathbb{N}$, let $\alpha_{i,j}^{(h)} \in \mathbb{K}^{M'_i}$ be the coefficient vector of the polynomial $X^h F_{i,j} \bmod P_i$, so that these vectors are given by

$$\alpha_{i,j}^{(0)} = \begin{bmatrix} F_{i,j}^{(0)} \\ \vdots \\ F_{i,j}^{(M'_i-1)} \end{bmatrix} \quad \text{and} \quad \alpha_{i,j}^{(h+1)} = \mathcal{C}_i \, \alpha_{i,j}^{(h)}.$$

Let then $A' = (A'_{i,j}) \in \mathbb{K}^{M' \times N'}$, where for every $i < \mu$ and $j < \nu$, the block $A'_{i,j} \in \mathbb{K}^{M'_i \times N'_j}$ is defined by

$$A'_{i,j} = \begin{bmatrix} \alpha_{i,j}^{(0)} & \cdots & \alpha_{i,j}^{(N'_j-1)} \end{bmatrix}.$$

Lemma 11. *A nonzero vector of $\mathbb{K}^{N'}$ is in the right nullspace of A' if and only if it is the coefficient vector of a solution $(Q_0, \ldots, Q_{\nu-1})$ to Problem 2.*

Proof. By definition $A'_{i,j}$ is the $M'_i \times N'_j$ matrix of the mapping $Q \mapsto QF_{i,j} \bmod P_i$, for Q in $\mathbb{K}[X]$ of degree less than N'_j. Thus, if $(Q_0, \ldots, Q_{\nu-1})$ is a ν-tuple of polynomials that satisfies the degree constraint (b) in Problem 2, applying A' to the coefficient vector of this tuple outputs the coefficients of the remainders $\sum_{0 \leqslant j < \nu} Q_j F_{i,j} \bmod P_i$, for $i = 0, \ldots, \mu - 1$. The claimed equivalence then follows immediately. □

The following lemma shows that A' possesses a Toeplitz-like structure, with displacement rank at most $\mu + \nu$. Together with Proposition 4 and Lemma 11, this gives our second proof of Theorem 1.

Lemma 12. *The displacement rank of A' with respect to $\Delta_{M',N'}$ is at most $\mu + \nu$. Furthermore, a corresponding generator of length $\mu+\nu$ for A' can be computed using $\mathcal{O}((\mu+\nu)\mathsf{M}(M'))$ operations in $\mathbb{K}$.*

Proof. We begin by constructing some matrices $Y \in \mathbb{K}^{M' \times (\mu+\nu)}$ and $Z \in \mathbb{K}^{(\mu+\nu) \times N'}$ such that $\Delta_{M',N'}(A')$ is equal to the product YZ. Define first the matrix

$$\mathcal{C} = \begin{bmatrix} \mathcal{C}_0 & 0 & \cdots & 0 \\ 0 & \mathcal{C}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{C}_{\mu-1} \end{bmatrix} \in \mathbb{K}^{M' \times M'}.$$

Up to μ columns, $\mathcal{C}$ coincides with $\mathcal{Z}_{M'}$; we make this explicit as follows. For i in $\{0, \ldots, \mu - 1\}$,

define

$$v_i = \begin{bmatrix} P_i^{(0)} \\ \vdots \\ P_i^{(M'_i-1)} \end{bmatrix} \in \mathbb{K}^{M'_i}, \qquad V_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ v_i \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{K}^{M'}, \qquad W_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{K}^{M'}, \tag{12}$$

where the last entry of v_i in V_i and the coefficient 1 in W_i have the same index, namely $M'_0 + \cdots + M'_i - 1$. (Hence the last vector $V_{\mu-1}$ only contains $v_{\mu-1}$, without a 1 after it.) Then, defining $V = [V_0 \cdots V_{\mu-1}] \in \mathbb{K}^{M' \times \mu}$ and $W = [W_0 \cdots W_{\mu-1}] \in \mathbb{K}^{M' \times \mu}$, we obtain

$$\mathcal{C} = \mathcal{Z}_{M'} - V_0 W_0^T - \cdots - V_{\mu-1} W_{\mu-1}^T \;=\; \mathcal{Z}_{M'} - V W^T.$$

As before, we use the convention that an indexed object is zero when the index is out of the allowed bounds for this object. For j in $\{0, \ldots, \nu-1\}$, let us further define

$$V'_j = \begin{bmatrix} \alpha_{0,j}^{(0)} \\ \vdots \\ \alpha_{\mu-1,j}^{(0)} \end{bmatrix} - \begin{bmatrix} \alpha_{0,j-1}^{(N'_{j-1})} \\ \vdots \\ \alpha_{\mu-1,j-1}^{(N'_{j-1})} \end{bmatrix} \in \mathbb{K}^{M'} \quad \text{and} \quad W'_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{K}^{N'}, \tag{13}$$

with the coefficient 1 in W'_j at index $N'_0 + \cdots + N'_{j-1}$, and the compound matrices

$$V' = [V'_0 \cdots V'_{\nu-1}] \in \mathbb{K}^{M' \times \nu} \quad \text{and} \quad W' = [W'_0 \cdots W'_{\nu-1}] \in \mathbb{K}^{N' \times \nu}.$$

Then, we claim that the matrices

$$Y = [\, -V \quad V' \,] \in \mathbb{K}^{M' \times (\mu+\nu)} \quad \text{and} \quad Z = \begin{bmatrix} W^T A' \mathcal{Z}_{N'}^T \\ W'^T \end{bmatrix} \in \mathbb{K}^{(\mu+\nu) \times N'} \tag{14}$$

are generators for A' for the Toeplitz-like displacement structure, *i.e.*, that

$$A' - \mathcal{Z}_{M'} A' \mathcal{Z}_{N'}^T = Y Z.$$

By construction, we have $\mathcal{C} A' = (B_{i,j})_{i<\mu, j<\nu} \in \mathbb{K}^{M' \times N'}$, with $B_{i,j}$ given by

$$B_{i,j} \;=\; \mathcal{C}_i A'_{i,j} \;=\; \begin{bmatrix} \alpha_{i,j}^{(1)} & \cdots & \alpha_{i,j}^{(N'_j-1)} & \alpha_{i,j}^{(N'_j)} \end{bmatrix} \in \mathbb{K}^{M'_i \times N'_j}.$$

As a consequence, $A' - \mathcal{C}\, A'\, \mathcal{Z}_{N'}^T = V'W'^T$, so finally we get, as claimed,

$$\begin{aligned} A' - \mathcal{Z}_{M'}\, A'\, \mathcal{Z}_{N'}^T &= A' - (\mathcal{C} + VW^T)A'\mathcal{Z}_{N'}^T \\ &= A' - \mathcal{C}\, A'\, \mathcal{Z}_{N'}^T - VW^T A'\mathcal{Z}_{N'}^T \\ &= V'W'^T - VW^T A\, \mathcal{Z}_{N'}^T \\ &= YZ. \end{aligned}$$

To compute Y and Z, the only non-trivial steps are those giving V' and W^TA'. For the former, we have to compute the coefficients of $X^{N'_j}F_{i,j} \bmod P_i$ for every $i < \mu$ and $j < \nu - 1$. For fixed i and j, this can be done using fast Euclidean division in $\mathcal{O}(\mathsf{M}(M'_i + N'_j))$ operations in $\mathbb{K}$, which is $\mathcal{O}(\mathsf{M}(M'_i) + \mathsf{M}(N'_j))$. Summing over the indices $i < \mu$ and $j < \nu - 1$, this gives a total cost of $\mathcal{O}(\nu\mathsf{M}(M') + \mu\mathsf{M}(N'))$ operations. This is $\mathcal{O}((\mu + \nu)\mathsf{M}(M'))$, since by assumption $N' \leqslant M' + 1$.

Finally, we show that W^TA' can be computed using $\mathcal{O}((\mu+\nu)\mathsf{M}(M'))$ operations as well. Computing this matrix amounts to computing the rows of A' of indices $M'_0 + \cdots + M'_i - 1$, for $i < \mu$. By construction of A', this means that we want to compute the coefficients of degree $M'_i - 1$ of $X^hF_{i,j} \bmod P_i$ for $h = 0, \ldots, N'_j - 1$ and for all i, j. Unfortunately, the naive approach leads to a cost proportional to $M'N'$ operations, which is not acceptable. However, for i and j fixed, Lemma 13 below shows how to do this computation using only $\mathcal{O}(\mathsf{M}(M'_i) + \mathsf{M}(N'_j))$ operations, which leads to the announced cost by summing over i and j. □

Lemma 13. *Let $P \in \mathbb{K}[X]$ be monic of degree m, let $F \in \mathbb{K}[X]$ be of degree less than m, and for $i \geqslant 0$ let c_i denote the coefficient of degree $m-1$ of $X^iF \bmod P$. For $n \geqslant 1$ we can compute $c_0, \ldots, c_{n-1}$ using $\mathcal{O}(\mathsf{M}(m) + \mathsf{M}(n))$ operations in $\mathbb{K}$.*

Proof. Writing $F = \sum_{0 \leqslant j < m} f_j X^j$ we have $X^iF \bmod P = \sum_{0 \leqslant j < m} f_j \big(X^{i+j} \bmod P\big)$. Hence $c_i = \sum_{0 \leqslant j < m} f_j b_{i+j}$, with b_i denoting the coefficient of degree $m-1$ of $X^i \bmod P$. Since $b_0 = \cdots = b_{m-2} = 0$ and $b_{m-1} = 1$, we can deduce $c_0, \ldots, c_{n-1}$ from $b_{m-1}, b_m, \ldots, b_{m+n-2}$ in time $\mathcal{O}(\mathsf{M}(n))$ by multiplication by the lower triangular Toeplitz matrix $[f_{m+j-i-1}]_{i,j}$ of order $n-1$.

Thus, we are left with the question of computing the $n-1$ coefficients $b_m, \ldots, b_{m+n-2}$. Writing P as $P = X^m + \sum_{0 \leqslant j < m} p_j X^j$ and using the fact that $X^iP \bmod P = 0$ for all $i \geqslant 0$, we see that the b_i's are generated by a linear recurrence of order m with constant coefficients:

$$b_{i+m} + \sum_{0 \leqslant j < m} p_j b_{i+j} = 0 \quad \text{for all } i \geqslant 0.$$

Consequently, $b_m, \ldots, b_{m+n-2}$ can be deduced from $b_0, \ldots, b_{m-1}$ in time $\mathcal{O}(\frac{n}{m}\mathsf{M}(m))$, which is $\mathcal{O}(\mathsf{M}(m) + \mathsf{M}(n))$, by $\lceil \frac{n-1}{m} \rceil$ calls to Shoup's algorithm for extending a linearly recurrent sequence [41, Theorem 3.1]. □

Algorithm 3. Solving Problem 2.

Input: positive integers $\mu,\ \nu,\ M'_0,\ldots,M'_{\mu-1},\ N'_0,\ldots,N'_{\nu-1}$ and polynomial tuples $\{(P_i,F_{i,0},\ldots,F_{i,\nu-1})\}_{0\leqslant i<\mu}$ in $\mathbb{K}[X]^{\nu+1}$ such that for all i, P_i is monic of degree M'_i and $\deg(F_{i,j})<M'_i$ for all j.

Output: polynomials $Q_0,\ldots,Q_{\nu-1}$ in $\mathbb{K}[X]$ such that (a), (b), (c).

1. Compute v_i and V_i for $i<\mu$, as defined in (12); compute $V=[V_0\ \cdots\ V_{\mu-1}]$

2. Compute W'_j for $j<\nu$, as defined in (13); compute $W'=[W'_0\ \cdots\ W'_{\nu-1}]$

3. Compute $\alpha_{i,j}^{(N'_j)}$, that is, the coefficients of $X^{N'_j}F_{i,j} \bmod P_i$, for $i<\mu$, $j<\nu-1$ (e.g. using fast Euclidean division)

4. Compute V'_j for $j<\mu$, as defined in (13); compute $V'=[V'_0\ \cdots\ V'_{\nu-1}]$

5. Compute the row of index $M'_0+\cdots+M'_i-1$ of A', for $i<\mu$, that is, the coefficient of degree M'_i-1 of $X^hF_{i,j} \bmod P_i$, for $h<N'_j$, $j<\nu$ (see the proof of Lemma 13 for fast computation).

6. Compute W^TA' whose row of index i is the row of index $M'_0+\cdots+M'_i-1$ of A'

7. Compute the generators Y and Z as defined in (14)

8. Use the algorithm of Proposition 4 with input Y and Z; if there is no solution then exit with no solution, otherwise find the coefficients of $Q_0,\ldots,Q_{\nu-1}$

9. Return $Q_0,\ldots,Q_{\nu-1}$

Appendix A. On assumption $\mathbf{H_1}$

In this appendix, we discuss the relevance of assumption $\mathbf{H_1}$ that was introduced previously for Problem 1. In the introduction, we did not make any assumption on m and ℓ, but we mentioned that assumption $\mathbf{H_1}$, that is, $m\leqslant\ell$ is mostly harmless. The following lemma substantiates this claim, by showing that the case $m\geqslant\ell$ can be reduced to the case $m=\ell$. As before, we denote by G the master polynomial $\prod_{1\leqslant i\leqslant n}(X-x_i)$.

Lemma 14. *Let $s,\ell,m,n,b,\boldsymbol{k}$ be parameters for Problem 1, and suppose that $m\geqslant\ell$. If $b\leqslant n(m-\ell)$ then this problem has no solution. Otherwise, its solutions are the polynomials of the form $Q=Q^\star\,G^{m-\ell}$ with $Q^\star$ a solution for the parameters $s,\ell,\ell,n,b-n(m-\ell),\boldsymbol{k}$.*

Proof. Assume a solution exists, say Q, and let $Q_i(X,\boldsymbol{Y})=Q(X+x_i,Y_1+y_{i,1},\ldots,Y_s+y_{i,s})$ for $i=1,\ldots,n$. Every monomial of Q_i has the form $X^h\boldsymbol{Y}^{\boldsymbol{j}}$ with $h\geqslant m-\ell$, since $|\boldsymbol{j}|\leqslant\ell$ by condition (ii) and $h+|\boldsymbol{j}|\geqslant m$ by condition (iv). Therefore, $X^{m-\ell}$ divides Q_i and, shifting back the coordinates for each i, we deduce that $G^{m-\ell}$ divides Q. It follows in particular

that $G(X)^{m-\ell}$ divides $Q(X, X^{k_1}Y_1, \ldots, X^{k_s}Y_s)$, which is nonzero by condition (i) and whose degree in X is less than b by condition (iii); since $G(X)^{m-\ell}$ has degree $n(m-\ell)$, this implies

$$n(m-\ell) < b.$$

Let us now consider $Q^\star = Q/G^{m-\ell}$ and show that it solves Problem 1 for the parameters $s, \ell, \ell, n, b-n(m-\ell), \boldsymbol{k}$. First, $Q^\star$ clearly satisfies conditions (i) and (ii). Furthermore,

$$Q^\star(X, X^{k_1}Y_1, \ldots, X^{k_s}Y_s) = Q(X, X^{k_1}Y_1, \ldots, X^{k_s}Y_s)/G(X)^{m-\ell},$$

so that condition (iii) holds with b replaced by $b-n(m-\ell)$. Finally, $Q^\star$ satisfies condition (iv) with m replaced by ℓ: writing $Q_i^\star(X, \boldsymbol{Y}) = Q^\star(X+x_i, Y_1+y_{i,1}, \ldots, Y_s+y_{i,s})$ for $i \in \{1, \ldots, n\}$, we have

$$Q_i^\star(X, \boldsymbol{Y}) = \frac{Q_i(X, \boldsymbol{Y})}{X^{m-\ell}\, G_i(X)^{m-\ell}}, \qquad G_i(X) = \prod_{h \neq i}(X + x_i - x_h);$$

all the monomials of $Q_i(X, \boldsymbol{Y})/X^{m-\ell}$ have the form $X^h \boldsymbol{Y}^{\boldsymbol{j}}$ with $h + |\boldsymbol{j}| \geqslant m - (m-\ell) = \ell$ and, since $G_i(0) \neq 0$, the same holds for $Q_i^\star(X, \boldsymbol{Y})$.

Conversely, let Q' be *any* solution to Problem 1 with parameters $s, \ell, \ell, n, b-n(m-\ell), \boldsymbol{k}$. Proceeding as in the previous paragraph, one easily verifies that the product $Q'\, G^{m-\ell}$ is a solution to Problem 1 with parameters $s, \ell, m, n, b, \boldsymbol{k}$, so the proof is complete. □

Appendix B. On a generalization of assumption $\mathbf{H_4}$

In this appendix, we show the relevance of the assumption "$k_i < n$ for some $i \in \{1, \cdots, n\}$" when considering Problem 1; in particular when we assume $\mathbf{H_2} : k_1 = \cdots = k_s =: k$, this shows the relevance of $\mathbf{H_4} : k < n$. Namely, when $k_i \geqslant n$ for every i, Lemma 15 below gives an explicit solution to Problem 1. Although the assumption $k < n$ is natural in the coding theory context, we provide here an argument to support the above mentioned more general assumption independently from any application context.

Lemma 15. *Let $s, \ell, m, n, b, \boldsymbol{k}$ be parameters for Problem 1 and suppose that $k_r \geqslant n$ for $r = 1, \ldots, s$. If $b \leqslant mn$ then this problem has no solution. Otherwise, a solution is given by the polynomial G^m (considered as an element of $\mathbb{K}[X, \boldsymbol{Y}]$).*

Proof. If $b > mn$ then it is easily checked that G^m satisfies conditions (i)–(iv) and thus solves Problem 1. Now, to conclude the proof, let us show that if Problem 1 admits a solution Q, then $b > mn$ must hold. Let $d = \deg_{\boldsymbol{Y}} Q$. If $d \geqslant m$, then the weighted-degree condition (iii) gives $b > \deg_X Q(X, X^{\boldsymbol{k}}\boldsymbol{Y}) \geqslant d(\min_r k_r) \geqslant mn$. Let us finally assume $d < m$. Following the proof of Lemma 14, we can write $Q = G(X)^{m-d}\, Q^\star$ for some $Q^\star$ in $\mathbb{K}[X, \boldsymbol{Y}]$ such that $\deg_{\boldsymbol{Y}} Q^\star = d$. Then, the weighted-degree condition gives $b > n(m-d) + \deg_X Q^\star(X, X^{\boldsymbol{k}}\boldsymbol{Y}) \geqslant n(m-d) + dn = mn$. □

Appendix C. Parameter constraints

We have seen that following the linear algebra point of view for solving Problem 1, the linearization of condition (iv) with the degree constraints (ii) and (iii) yields a homogeneous linear system with M equations and N unknowns. Then, one may assume $M < N$ to ensure the existence of a solution to the problem. In particular in the coding theory context, this gives a constraint on the input parameters which is usually used to choose s, m, ℓ with respect to n, k, b.

Our goal in this appendix is to verify that in the coding theory context and considering the polynomial lattice point of view, the constraint on the parameters which ensures the existence of a nontrivial solution to Problem 1 is the same inequality $M < N$. In other words, if we require such a guarantee on the existence of a solution, these two approaches can solve Problem 1 for exactly the same set of parameters.

Let us assume $\mathbf{H_1}, \mathbf{H_2}, \mathbf{H_3}, \mathbf{H_4}$ and recall that in this context we have the identities

$$\Gamma = \{\boldsymbol{j} \in \mathbb{N}^s : |\boldsymbol{j}| \leqslant \ell\} \quad \text{and} \quad |\Gamma| = \binom{s+\ell}{s},$$

$$N \;=\; \sum_{|\boldsymbol{j}|\leqslant \ell} N_{\boldsymbol{j}} \;=\; \sum_{|\boldsymbol{j}|\leqslant \ell} b - |\boldsymbol{j}|k \;=\; \binom{s+\ell}{s} b \;-\; \sum_{|\boldsymbol{j}|\leqslant \ell} |\boldsymbol{j}|k,$$

$$M \;=\; \sum_{|\boldsymbol{i}|< m} M_{\boldsymbol{i}} \;=\; \sum_{|\boldsymbol{i}|< m} n(m - |\boldsymbol{i}|) \;=\; \binom{s+m}{s+1} n.$$

In the polynomial lattice based approach (see for instance [12]), the list-size and vanishing conditions (ii) and (iv) are consequences of belonging to the lattice $\mathcal{L}$ (two possibilities for building $\mathcal{L}$ are presented in Section 1). Thus, remembering how we built $\mathcal{L}$ with basis polynomials $h(X, X^k\boldsymbol{Y})$ to account for the weighted-degree $(1, k, \ldots, k)$, we note that a shortest (nonzero) vector in the lattice $\mathcal{L}$ will correspond to a polynomial $Q(X, Y)$ satisfying (ii) and (iv) such that $\deg_X Q(X, X^k\boldsymbol{Y})$ is minimal.

Besides, it is known [12, Section 2.2] that any shortest vector in $\mathcal{L}$ has degree at most $\deg(\det \mathcal{L}) \;/\; \dim \mathcal{L}$; that is, the corresponding $Q(X, \boldsymbol{Y})$ will satisfy $\deg_X Q(X, X^k\boldsymbol{Y}) \leqslant \deg(\det \mathcal{L}) \;/\; \dim \mathcal{L}$. Then, in order to guarantee that a shortest vector in $\mathcal{L}$ will correspond to a polynomial $Q(X, \boldsymbol{Y})$ that satisfies the weighted-degree condition (iii), we rely on the following condition:

$$\frac{\deg(\det \mathcal{L})}{\dim \mathcal{L}} \;<\; b. \tag{15}$$

For both lattices described in Section 1, one can verify that

$$\det \mathcal{L} \;=\; G(X)^{i_1} X^{i_2}, \qquad i_1 = \sum_{0\leqslant|\boldsymbol{i}|<m} n(m - |\boldsymbol{i}|), \quad i_2 = \sum_{0\leqslant|\boldsymbol{j}|\leqslant\ell} |\boldsymbol{j}|k$$

so that the condition (15) can be rewritten

$$\frac{1}{\binom{s+\ell}{s}} \left(\sum_{0\leqslant|\boldsymbol{i}|<m} n(m - |\boldsymbol{i}|) \;+\; \sum_{0\leqslant|\boldsymbol{j}|\leqslant\ell} |\boldsymbol{j}|k \right) \;<\; b,$$

which is precisely the assumption $M < N$ seen before.

Thus, in both approaches, the same assumption can be used to ensure that we will find a solution to our problem. Now, solving Problem 1 without this assumption on the parameters, as we presented using structured linear algebra, can also be done in the context of polynomial lattice based algorithms. Indeed, even if the inequality (15) does not hold, one can still build the lattice, compute a shortest vector therein and check whether the corresponding polynomial Q satisfies the weighted-degree condition. If it does, then Q is a solution to our problem; otherwise, there is no solution with these parameters.

Acknowledgments. Muhammad F. I. Chowdhury and Éric Schost were supported by NSERC and by the Canada Research Chairs program. We thank the three reviewers for their helpful comments on a preliminary version of this work [11], and especially the second one for suggesting a shorter proof of Lemma 13.

References

[1] M. Alekhnovich. Linear diophantine equations over polynomials and soft decoding of Reed-Solomon codes. *IEEE Transactions on Information Theory*, 51(7):2257–2265, July 2005.

[2] B. Beckermann. A reliable method for computing M-Padé approximants on arbitrary staircases. *Journal of Computational and Applied Mathematics*, 40(1):19–42, 1992.

[3] B. Beckermann and G. Labahn. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM Journal on Matrix Analysis and Applications*, 15(3):804–823, July 1994.

[4] P. Beelen and K. Brander. Key equations for list decoding of Reed-Solomon codes and how to solve them. *Journal of Symbolic Computation*, 45(7):773–786, 2010.

[5] D. J. Bernstein. Simplified high-speed high-distance list decoding for alternant codes. In *PQCrypto'11*, pages 200–216, Berlin, Heidelberg, 2011. Springer-Verlag.

[6] R. R. Bitmead and B. D. O. Anderson. Asymptotically fast solution of Toeplitz and related systems of linear equations. *Linear Algebra and its Applications*, 34:103–116, 1980.

[7] A. Bostan, C.-P. Jeannerod, and É. Schost. Solving structured linear systems with large displacement rank. *Theoretical Computer Science*, 407(1-3):155–181, November 2008.

[8] K. Brander. *Interpolation and List Decoding of Algebraic Codes*. PhD thesis, Technical University of Denmark, 2010.

[9] P. Busse. *Multivariate List Decoding of Evaluation Codes with a Gröbner Basis Perspective*. PhD thesis, University of Kentucky, 2008.

[10] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[11] M. F. I. Chowdhury, C.-P. Jeannerod, V. Neiger, É. Schost, and G. Villard. On the complexity of multivariate interpolation with multiplicities and of simultaneous polynomial approximations. Presented at the 10th Asian Symposium on Computer Mathematics (ASCM), Beijing, China, October 2012.

[12] H. Cohn and N. Heninger. Ideal forms of Coppersmith's theorem and Guruswami-Sudan list decoding. In Bernard Chazelle, editor, *Innovations in Computer Science*, pages 298–308. Tsinghua University Press, 2011. Extended version available at http://arxiv.org/pdf/1008.1284.

[13] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comp.*, 9(3):251–280, 1990.

[14] R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978.

[15] G. L. Feng and K. K. Tzeng. A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes. *IEEE Transactions on Information Theory*, 37(5):1274–1287, 1991.

[16] P. Gaborit and O. Ruatta. Improved Hermite multivariate polynomial interpolation. In *2006 IEEE International Symposium on Information Theory*, pages 143–147, 2006.

[17] François Le Gall. Powers of tensors and fast matrix multiplication, 2014. http://arxiv.org/abs/1401.7714.

[18] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra (second edition)*. Cambridge University Press, 2003.

[19] P. Giorgi, C.-P. Jeannerod, and G. Villard. On the complexity of polynomial matrix computations. In *ISSAC'03*, pages 135–142, New York, NY, USA, 2003. ACM.

[20] S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $K[x]$. *Journal of Symbolic Computation*, 47(4):422–453, April 2012.

[21] V. Guruswami and A. Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008.

[22] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[23] H. Hasse. Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenem Konstantenkörper bei beliebiger Charakteristik. *Journal für die reine und angewandte Mathematik*, 175:50–54, 1936.

[24] E. Kaltofen. Asymptotically fast solution of Toeplitz-like singular linear systems. In *ISSAC'94*, pages 297–304. ACM, 1994.

[25] E. Kaltofen and D. Saunders. On Wiedemann's method of solving sparse linear systems. In *AAECC-9*, volume 539 of *Lecture Notes in Computer Science*, pages 29–38. Springer Verlag, 1991.

[26] R. Kötter. Fast generalized minimum-distance decoding of algebraic-geometry and Reed-Solomon codes. *IEEE Transactions on Information Theory*, 42(3):721 –737, May 1996.

[27] K. Lee and M. E. O'Sullivan. List decoding of Reed-Solomon codes from a Gröbner basis perspective. *Journal of Symbolic Computation*, 43(9):645 – 658, 2008.

[28] R. J. McEliece. The Guruswami-Sudan decoding algorithm for Reed-Solomon codes, 2003. IPN Progress Report 42-153.

[29] H. M. Möller and B. Buchberger. The construction of multivariate polynomials with preassigned zeros. In *EUROCAM'82*, volume 144 of *Lecture Notes in Computer Science*, pages 24–31. Springer, 1982.

[30] M. Morf. Doubling algorithms for Toeplitz and related equations. *IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 954–959, 1980.

[31] J. S. R. Nielsen. *List Decoding of Algebraic Codes*. PhD thesis, Technical University of Denmark, 2013.

[32] R. R. Nielsen and T. Høholdt. Decoding Reed-Solomon codes beyond half the minimum distance. In *Coding Theory, Cryptography and Related Areas*, pages 221–236. Springer-Verlag, 2000.

[33] V. Olshevsky and M. A. Shokrollahi. A displacement approach to efficient decoding of algebraic-geometric codes. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC'99, pages 235–244, New York, NY, USA, 1999. ACM.

[34] V. Y. Pan. *Structured Matrices and Polynomials*. Birkhäuser Boston Inc., 2001.

[35] F. Parvaresh and A. Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *FOCS'05*, pages 285–294, 2005.

[36] J.-R. Reinhard. Algorithme LLL polynomial et applications. Master's thesis, École Polytechnique, Paris, France, 2003.

[37] R. M. Roth. *Introduction to Coding Theory*. Cambridge University Press, 2007.

[38] R. M. Roth and G. Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, 46(1):246–257, January 2000.

[39] S. Sarkar and A. Storjohann. Normalization of row reduced matrices. In *ISSAC'11*, pages 297–304, 2011.

[40] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, October 1980.

[41] V. Shoup. A fast deterministic algorithm for factoring polynomials over finite fields of small characteristic. In *ISSAC'91*, pages 14–21. ACM, 1991.

[42] A. Storjohann. Notes on computing minimal approximant bases. In *Challenges in Symbolic Computation Software*, Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2006.

[43] A. Stothers. *On the Complexity of Matrix Multiplication*. PhD thesis, University of Edinburgh, 2010.

[44] M. Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, March 1997.

[45] P. V. Trifonov. Efficient interpolation in the Guruswami-Sudan algorithm. *IEEE Transactions on Information Theory*, 56(9):4341–4349, September 2010.

[46] V. Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th symposium on Theory of Computing*, STOC'12, pages 887–898. ACM, 2012.

[47] A. Zeh, C. Gentner, and D. Augot. An interpolation procedure for list decoding Reed-Solomon codes based on generalized key equations. *IEEE Transactions on Information Theory*, 57(9):5946–5959, September 2011.

[48] R. Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM'79*, volume 72 of *Lecture Notes in Computer Science*. Springer Verlag, 1979.