Faster Algorithms for Multivariate Interpolation with Multiplicities and Simultaneous Polynomial Approximations

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

***Abstract*—The interpolation step in the Guruswami-Sudan algorithm is a bivariate interpolation problem with multiplicities commonly solved in the literature using either structured linear algebra or basis reduction of polynomial lattices. This problem has been extended to three or more variables; for this generalization, all fast algorithms proposed so far rely on the lattice approach. In this paper, we reduce this multivariate interpolation problem to a problem of simultaneous polynomial approximations, which we solve using fast 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, and folded Reed-Solomon codes. In particular, for Reed-Solomon list-decoding with re-encoding, our approach has complexity $\mathcal{O}\tilde{~}(\ell^{\omega-1}m^2(n-k))$, where ℓ, m, n, k are the list size, the multiplicity, the number of sample points and the dimension of the code, and ω is the exponent of linear algebra; this accelerates the previously fastest known algorithm by a factor of ℓ/m.**

***Index Terms*—multivariate polynomial interpolation, polynomial approximation, structured matrix, list decoding, Reed-Solomon codes.**

I. Introduction

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, n, b are respectively known as the *number of variables*, *list size*, *code length*, and as an *agreement parameter*. The parameters $m_1, \ldots, m_n$ are known as *multiplicities* associated with each of the n points; furthermore, the s variables are associated with some *weights* $k_1, \ldots, k_s$. In the application to list-decoding of Reed-Solomon codes, we have $s = 1$, all the multiplicities are equal to a same value m, $n-b/m$ is an upper bound on the number of errors allowed on a received word, and the weight $k := k_1$ is such that $k+1$ is the *dimension* of the code. Further details concerning the applications of our results to list-decoding and soft-decoding of Reed-Solomon codes are given in Section IV.

We stress that here we do not address the issue of choosing the parameters $s, \ell, m_1, \ldots, m_n$ 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, $\mathbb{Z}$ is the set of integers, $\mathbb{Z}_{\geqslant 0}$ the set of nonnegative integers, and $\mathbb{Z}_{>0}$ the set of positive integers. Besides, $\deg_{Y_1,\ldots,Y_s}$ denotes the total degree with respect to the variables $Y_1, \ldots, Y_s$, and $\mathrm{wdeg}_{k_1,\ldots,k_s}$ denotes the weighted-degree with respect to weights $k_1, \ldots, k_s \in \mathbb{Z}$ on variables $Y_1, \ldots, Y_s$, respectively; that is, for a polynomial $Q = \sum_{(j_1,\ldots,j_s)} Q_{j_1,\ldots,j_s}(X) Y_1^{j_1} \cdots Y_s^{j_s}$,

$$\mathrm{wdeg}_{k_1,\ldots,k_s}(Q) = \max_{j_1,\ldots,j_s} \left(\deg(Q_{j_1,\ldots,j_s}) + j_1k_1 + \cdots + j_sk_s\right).$$

Problem 1. Multivariate Interpolation

Input: $s, \ell, n, m_1, \ldots, m_n$ in $\mathbb{Z}_{>0}$, $b, k_1, \ldots, k_s$ in $\mathbb{Z}$ 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 pairwise distinct.

Output: a polynomial Q in $\mathbb{K}[X, Y_1, \ldots, Y_s]$ such that
- *(i)* Q is nonzero,
- *(ii)* $\deg_{Y_1,\ldots,Y_s}(Q) \leqslant \ell$,
- *(iii)* $\mathrm{wdeg}_{k_1,\ldots,k_s}(Q) < b$,
- *(iv)* $Q(x_i, y_{i,1}, \ldots, y_{i,s}) = 0$ with multiplicity at least m_i for $1 \leqslant i \leqslant n$.

We call conditions *(ii)*, *(iii)*, and *(iv)* the *list-size* condition, the *weighted-degree* condition, and the *vanishing* condition, respectively. Note that a point $(x, y_1, \ldots, y_s)$ is a zero of Q of *multiplicity at least* m if the shifted polynomial $Q(X+x, Y_1+y_1, \ldots, Y_s+y_s)$ has no monomial of total degree less than m; in characteristic zero or larger than m, this is equivalent to requiring that all the derivatives of Q of order up to $m-1$ vanish at $(x, y_1, \ldots, y_s)$.

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

Muhammad F. I. Chowdhury is with Computer Science Department, University of Western Ontario, London ON, Canada.

Claude-Pierre Jeannerod is with Inria, Laboratoire LIP (CNRS, ENS de Lyon, Inria, UCBL), Université de Lyon, France.

Vincent Neiger is with ENS de Lyon, Laboratoire LIP (CNRS, ENS de Lyon, Inria, UCBL), Université de Lyon, France and Computer Science Department, University of Western Ontario, London ON, Canada.

Éric Schost is with Computer Science Department, University of Western Ontario, London ON, Canada.

Gilles Villard is with CNRS, Laboratoire LIP (CNRS, ENS de Lyon, Inria, UCBL), Université de Lyon, France.

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.

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, n, $m_1, \ldots, m_n$, while the number N of unknowns derives from conditions *(ii)* and *(iii)* and thus depends on s, ℓ, b, $k_1, \ldots, k_s$. 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 of the interpolation step of the Guruswami-Sudan algorithm [49], [23] to s variables $Y_1, \ldots, Y_s$, distinct multiplicities, and distinct weights. The multivariate case $s > 1$ occurs for instance in Parvaresh-Vardy codes [40] or folded Reed-Solomon codes [22]. Distinct multiplicities occur for instance in the interpolation step in soft-decoding of Reed-Solomon codes [28]. We note that this last problem is different from our context since the x_i are not necessarily pairwise distinct; we briefly explain in Section IV-D how to deal with this case.

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. Simultaneous Polynomial Approximations

Input: μ, ν, $M'_0, \ldots, M'_{\mu-1}$, $N'_0, \ldots, N'_{\nu-1}$ in $\mathbb{Z}_{>0}$ and tuples $\{(P_i, F_{i,0}, \ldots, F_{i,\nu-1})\}_{0 \leqslant i < \mu}$ of polynomials in $\mathbb{K}[X]$ 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)* the Q_j are not all zero,
- *(b)* for $0 \leqslant j < \nu$, $\deg(Q_j) < N'_j$,
- *(c)* for $0 \leqslant i < \mu$, $\sum_{0 \leqslant j < \nu} F_{i,j} Q_j = 0 \bmod P_i$.

Main complexity results and applications. We first show in Section II how to reduce Problem 1 to Problem 2 efficiently via a generalization of the techniques introduced by Zeh, Gentner, and Augot [54] and Zeh [53, Section 5.1.1] for, respectively, the list-decoding and soft-decoding of Reed-Solomon codes.

Then, in Section III we present two algorithms for solving Problem 2. Each of them involves a linearization of the univariate equations *(c)* into a specific 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 both systems have M' equations in N' unknowns. (As for our first problem, we need not assume that $M' < N'$.) Furthermore, the structure of these systems allows us to solve them efficiently using the algorithm of Bostan, Jeannerod, and Schost in [8].

Our first algorithm, detailed in Section III-B, solves Problem 2 by following the derivation of so-called *extended key equations* (EKE), initially introduced for the particular case of Problem 1 by Roth and Ruckenstein [43] when $s = m = 1$ and then by Zeh, Gentner, and Augot [54] when $s = 1$ and $m \geqslant 1$; the matrix of the system is mosaic-Hankel. In our second algorithm, detailed in Section III-C, the linear system is more directly obtained from condition *(c)*, without resorting to EKEs, and has Toeplitz-like structure.

Both points of view lead to the same complexity result, stated in Theorem 2 below, 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 $\mathcal{O}\tilde{~}(\,)$ 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 $\omega < 2.38$ [15], [48], [51], [31]. 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 satisfies the super-linearity properties of [19, Ch. 8]. It follows from the algorithm of Cantor and Kaltofen [11] that $\mathsf{M}(d)$ can be taken in $\mathcal{O}(d \log(d) \log\log(d)) \subseteq \mathcal{O}\tilde{~}(d)$.

Combining Theorem 2 below with the above-mentioned reduction from Problem 1 to Problem 2, we immediately deduce the following cost bound for Problem 1.

Theorem 1. *Let*

$$\Gamma = \Big\{(j_1, \ldots, j_s) \in \mathbb{Z}^s_{\geqslant 0} \;\Big|\; j_1 + \cdots + j_s \leqslant \ell \\ \text{and } j_1 k_1 + \cdots + j_s k_s < b\Big\},$$

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

$$\mathcal{O}\big(\varrho^{\omega-1} \mathsf{M}(M) \log(M)^2\big) \subseteq \mathcal{O}\tilde{~}(\varrho^{\omega-1} M)$$

operations in $\mathbb{K}$. *This can be achieved using Algorithm 1 in Section II followed by Algorithm 2 or 3 in Section III. 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* $1/2$.

We will often refer to the two following assumptions on the input parameters:

H₁: $m \leqslant \ell$,
H₂: $b > 0$ and $b > \ell \cdot \max_{1 \leqslant j \leqslant s} k_j$.

Regarding **H₁**, we prove in Appendix A that the case $m > \ell$ can be reduced to the case $m = \ell$, so that this assumption can be made without loss of generality. Besides, it is easily verified that **H₂** is equivalent to having $\Gamma = \{(j_1, \ldots, j_s) \in \mathbb{Z}^s_{\geqslant 0} \mid j_1 + \cdots + j_s \leqslant \ell\}$; when $k_j > 0$ for some j, **H₂** means that we do not take ℓ uselessly large. Then, assuming **H₁** and **H₂**, we have $\varrho = |\Gamma| = \binom{s+\ell}{s}$.

As we will show in Section IV, in the context of the list-decoding of Reed-Solomon codes, applications of Theorem 1 include the interpolation step of the Guruswami-Sudan algorithm [23] in $\mathcal{O}\tilde{~}(\ell^{\omega-1} m_{\mathrm{GS}}^2 n)$ operations and the interpolation step of the Wu algorithm [52] in $\mathcal{O}\tilde{~}(\ell^{\omega-1} m_{\mathrm{Wu}}^2 n)$ operations, where m_{GS} and m_{Wu} are the respective multiplicities used in those algorithms; our result can also be adapted to the context of soft-decoding [28]. Besides, the re-encoding technique of Koetter and Vardy [29] can be used in conjunction with our algorithm in order to reduce the cost of the interpolation step of the Guruswami-Sudan algorithm to $\mathcal{O}\tilde{~}(\ell^{\omega-1} m_{\mathrm{GS}}^2 (n-k))$ operations.

In Theorem 1, 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 field extension $\mathbb{L}$ of degree $d \in \mathcal{O}(\log_{|\mathbb{K}|}(M))$, up to a cost increase by a factor in $\mathcal{O}(\mathsf{M}(d)\log(d))$.

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 d; such an f can be set up using an expected number of $\mathcal{O}\tilde{~}(d^2) \subseteq \mathcal{O}(M)$ operations in $\mathbb{K}$ [19, §14.9]. Then we solve Problem 1 over $\mathbb{L}$ by means of the algorithm of Theorem 1, thus using $\mathcal{O}\big(\varrho^{\omega-1}\mathsf{M}(M)\log(M)^2 \cdot \mathsf{M}(d)\log(d)\big)$ operations in $\mathbb{K}$. Finally, from this solution over $\mathbb{L}$ one can deduce a solution over $\mathbb{K}$ using $\mathcal{O}(Md)$ operations in $\mathbb{K}$. This last point comes from the fact that, as we shall see later in the paper, Problem 1 amounts to finding a nonzero vector u over $\mathbb{K}$ such that $Au = 0$ for some $M \times (M+1)$ matrix A over $\mathbb{K}$: once we have obtained a solution $\overline{u}$ over $\mathbb{L}$, it thus suffices to rewrite it as $\overline{u} = \sum_{0 \leqslant i < d} u_i X^i \neq 0$ and, noting that $Au_i = 0$ for all i, to find a nonzero u_i in $\mathcal{O}(Md)$ comparisons with zero and return it as a solution over $\mathbb{K}$.

Furthermore, 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 all the applications to error-correcting codes we consider in this paper, M is polynomial in n so that we can take $d = \mathcal{O}(1)$, and in those cases the cost bound in Theorem 1 holds for any field.

As said before, Theorem 1 relies on an efficient solution to Problem 2, which we summarize in the following theorem.

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

$$\mathcal{O}\big(\rho^{\omega-1}\mathsf{M}(M')\log(M')^2\big) \subseteq \mathcal{O}\tilde{~}(\rho^{\omega-1}M')$$

operations in $\mathbb{K}$. Algorithms 2 and 3 in Section III 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 $1/2$.

If $\mathbb{K}$ has fewer than $6(M'+1)^2$ elements, the remarks made after Theorem 1 still apply here.

Comparison with previous work. In the context of coding theory, most previous results regarding Problem 1 focus on the list-decoding of Reed-Solomon codes via the Guruswami-Sudan algorithm, in which $s = 1$ and the assumptions $\mathbf{H_1}$ and $\mathbf{H_2}$ are satisfied as well as

$\mathbf{H_3}$: $0 \leqslant k < n$ where $k := k_1$,
$\mathbf{H_4}$: $m_1 = \cdots = m_n = m$.

The assumption $\mathbf{H_3}$ corresponds to the coding theory context, where $k+1$ is the dimension of the code; then $k+1$ must be positive and at most n (the length of the received word). To support this assumption independently from any application context, we show in Appendix B that if $k \geqslant n$, then Problem 1 has either a trivial solution or no solution at all.

Previous results focus mostly on the Guruswami-Sudan case ($s = 1, m \geqslant 1$) and some of them more specifically on the Sudan case ($s = m = 1$); we summarize these results in Table I. In some cases [41], [1], [6], [13], 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 $\mathbf{H_1}$, the second most important parameter is ℓ, followed by m. In particular, our result in Section IV, Corollary 14 compares favorably to the cost $\mathcal{O}\tilde{~}(\ell^{\omega} mn)$ obtained by Cohn and Heninger [13] 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 [23], and $\ell = \mathcal{O}(n^{1/2}k^{-1/2})$ in the Sudan case [49]. With these parameter choices, our algorithms present a speedup $(n/k)^{1/2}$ over the algorithm in [13].

Most previous algorithms rely on linear algebra, either over $\mathbb{K}$ or over $\mathbb{K}[X]$. When 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 [30], [37], [33] are particular cases of the Buchberger-Möller algorithm [34], while Alekhnovich's algorithm [1] is a divide-and-conquer change of ordering algorithm for bivariate ideals.

Yet another line of work [43], [54] uses Feng and Tzeng's linear system solver [17], 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 III-B, but we will rely on the structured linear system solver of [8] in order to prove our main results. Prior to our work, Olshevsky and Shokrollahi also used structured linear algebra techniques [38], but it is unclear to us whether their encoding of the problem could lead to similar results as ours.

As said above, another approach rephrases the problem of computing Q in terms of polynomial matrix computations, that is, as linear algebra over $\mathbb{K}[X]$. Starting from known generators of the finitely generated $\mathbb{K}[X]$-module (or polynomial lattice) formed by solutions to Problem 1, the algorithms in [41], [32], [10], [4], [9], [6], [13] compute a Gröbner basis of this module (or a reduced lattice basis), in order to find a short vector therein. To achieve quasi-linear time in n, the algorithms in [4], [9] use a basis reduction subroutine due to Alekhnovich [1], while those in [6], [13] rely on a faster, randomized algorithm due to Giorgi, Jeannerod, and Villard [20].

This approach based on the computation of a reduced lattice basis was in particular the basis of the extensions to the multi-

TABLE I
COMPARISON OF OUR COSTS WITH PREVIOUS ONES FOR $s = 1$.

Sudan case ($m = 1$)		
Sudan [49]	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
Roth-Ruckenstein [43]	$\mathcal{O}(\ell n^2)$	$\mathcal{O}(n^{2+1/2}k^{-1/2})$
Olshevsky-Shokrollahi [38]	$\mathcal{O}(\ell n^2)$	$\mathcal{O}(n^{2+1/2}k^{-1/2})$
This paper	$\mathcal{O}(\ell^{\omega-1}\mathsf{M}(n)\log(n)^2)$	$\mathcal{O}\tilde{}(n^{\omega/2+1/2}k^{1/2-\omega/2})$
Guruswami-Sudan case ($m \geqslant 1$)		
Guruswami-Sudan [23]	$\mathcal{O}(m^6n^3)$	$\mathcal{O}(n^9k^6)$
Olshevsky-Shokrollahi [38]	$\mathcal{O}(\ell m^4n^2)$	$\mathcal{O}(n^{7+1/2}k^{4+1/2})$
Zeh-Gentner-Augot [54]	$\mathcal{O}(\ell m^4n^2)$	$\mathcal{O}(n^{7+1/2}k^{4+1/2})$
Kötter / McEliece [30], [33]	$\mathcal{O}(\ell m^4n^2)$	$\mathcal{O}(n^{7+1/2}k^{4+1/2})$
Reinhard [41]	$\mathcal{O}(\ell^3m^2n^2)$	$\mathcal{O}(n^{8+1/2}k^{3+1/2})$
Lee-O'Sullivan [32]	$\mathcal{O}(\ell^4mn^2)$	$\mathcal{O}(n^9k^3)$
Trifonov [50] (heuristic)	$\mathcal{O}(m^3n^2)$	$\mathcal{O}(n^5k^3)$
Alekhnovich [1]	$\mathcal{O}(\ell^4m^4\mathsf{M}(n)\log(n))$	$\mathcal{O}\tilde{}(n^{11}k^6)$
Beelen-Brander [4]	$\mathcal{O}(\ell^3\mathsf{M}(\ell mn)\log(n))$	$\mathcal{O}\tilde{}(n^8k^3)$
Bernstein [6]	$\mathcal{O}(\ell^\omega\mathsf{M}(\ell n)\log(n))$	$\mathcal{O}\tilde{}(n^{3\omega/2+5/2}k^{\omega/2+1/2})$
Cohn-Heninger [13]	$\mathcal{O}(\ell^\omega\mathsf{M}(mn)\log(n))$	$\mathcal{O}\tilde{}(n^{3\omega/2+2}k^{\omega/2+1})$
This paper	$\mathcal{O}(\ell^{\omega-1}\mathsf{M}(m^2n)\log(n)^2)$	$\mathcal{O}\tilde{}(n^{3\omega/2+3/2}k^{\omega/2+3/2})$

variate case $s > 1$ in [10], [9], [14]. In the multivariate case as well, the result in Theorem 1 improves on the best previously known bounds [10], [9], [14]; we detail those bounds and we prove this claim in Appendix C. In [18], the authors solve a problem similar to Problem 1 except that they do not assume that the x_i are distinct. For simple roots and under some genericity assumption on the points $\{(x_i, y_{i,1}, \dots, y_{i,s})\}_{1\leqslant i\leqslant n}$, this algorithm uses $O(n^{2+1/s})$ 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 ($m > 1$).

Regarding Problem 2, several particular cases of it are well-known. When all P_i 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], [47]. 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 [36, 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 [20], 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 2 compares favorably.

Outline of the paper. First, we show in Section II how to reduce Problem 1 to Problem 2; this reduction is essentially based on Lemma 4, which extends to the multivariate case $s > 1$ the results in [54], [53]. Then, after a reminder on algorithms for structured linear systems in Section III-A, we give two algorithms that both prove Theorem 2, in Sections III-B and III-C, respectively. The linearization in the first algorithm extends the derivation of extended key equations presented in [54] to the more general context of Problem 2, ending up with a mosaic-Hankel system. The second algorithm gives an alternative approach, in which the linearization is more straightforward and the structure of the matrix of the system is Toeplitz-like. We conclude in Section IV by presenting several applications to the list-decoding of Reed-Solomon codes, namely the Guruswami-Sudan algorithm, the re-encoding technique and the Wu algorithm, and by sketching how to adapt our approach to the soft-decoding of Reed-Solomon codes. Readers who are mainly interested in those applications may skip Section III, which contains the proofs of Theorems 1 and 2, and go directly to Section IV.

II. 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 gives an overview of this reduction. The main technical ingredient, stated in Lemma 4 below, generalizes to any $s \geqslant 1$ and (possibly) distinct multiplicities the result given for $s = 1$ by Zeh, Gentner, and Augot in [54, Proposition 3]. To prove it, we use the same steps as in [54]; we rely on the notion of Hasse derivatives, which allows us to write Taylor expansions in positive characteristic (see Hasse [24] or Roth [42, pp. 87, 276]).

For simplicity, in the rest of this paper we will use boldface letters to denote s-tuples of objects: $\boldsymbol{Y} = (Y_1, \dots, Y_s)$, $\boldsymbol{k} = (k_1, \dots, k_s)$, etc. In the special case of s-tuples of integers, we also write $|\boldsymbol{k}| = k_1 + \dots + k_s$, and comparison and addition of multi-indices in $\mathbb{Z}^s_{\geqslant 0}$ are defined componentwise. For example, writing $\boldsymbol{i} \leqslant \boldsymbol{j}$ is equivalent to $i_1 \leqslant j_1, \dots, i_s \leqslant j_s$, and $\boldsymbol{i} - \boldsymbol{j}$ denotes $(i_1 - j_1, \dots, i_s - j_s)$. If $\boldsymbol{y} = (y_1, \dots, y_s)$ is in $\mathbb{K}[X]^s$ and $\boldsymbol{i} = (i_1, \dots, i_s)$ is in $\mathbb{Z}^s_{\geqslant 0}$, then $\boldsymbol{Y} - \boldsymbol{y} = (Y_1 - y_1, \dots, Y_s - y_s)$ 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 $Y_1, \dots, Y_s$ 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 $\mathbb{Z}_{\geqslant 0}^s$, 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 help rephrase the vanishing condition *(iv)* of Problem 1 for one of the points $\{(x_r, \boldsymbol{y}_r)\}_{1 \leqslant r \leqslant n}$.

Lemma 3. *Let $(x, y_1, \ldots, y_s)$ be a point in $\mathbb{K}^{s+1}$ and $\boldsymbol{R} = (R_1, \ldots, R_s)$ in $\mathbb{K}[X]^s$ be such that $R_j(x) = y_j$ for $1 \leqslant j \leqslant s$. Then, for any polynomial Q in $\mathbb{K}[X, \boldsymbol{Y}]$, $Q(x, \boldsymbol{y}) = 0$ with multiplicity at least m if and only if for all $\boldsymbol{i}$ in $\mathbb{Z}_{\geqslant 0}^s$ such that $|\boldsymbol{i}| < m$,*

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

Proof. Up to a shift, one can assume that the point is $(x, y_1, \ldots, y_s) = (0, \mathbf{0})$; in other words, it suffices to show that for $\boldsymbol{R}(0) = \mathbf{0} \in \mathbb{K}^s$, we have $Q(0, \mathbf{0}) = 0$ with multiplicity at least m if and only if, for all $\boldsymbol{i}$ in $\mathbb{Z}_{\geqslant 0}^s$ such that $|\boldsymbol{i}| < m$, $X^{m-|\boldsymbol{i}|}$ divides $Q^{[\boldsymbol{i}]}(X, \boldsymbol{R})$.

Assume first that $(0, \mathbf{0}) \in \mathbb{K}^{s+1}$ 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, $\boldsymbol{R}(0) = \mathbf{0} \in \mathbb{K}^s$ 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 $\mathbb{Z}_{\geqslant 0}^s$ 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 $|\boldsymbol{i}| < m$. □

Thus, for each of the points $\{(x_r, \boldsymbol{y}_r)\}_{1 \leqslant r \leqslant n}$ in Problem 1, such a rewriting of the vanishing condition *(iv)* for this point holds. Now intervenes the fact that the x_i are distinct: the polynomials $(X - x_a)^\alpha$ and $(X - x_b)^\beta$ are coprime for $a \neq b$, so that simultaneous divisibility by both those polynomials is equivalent to divisibility by their product $(X-x_a)^\alpha (X-x_b)^\beta$. Using the s-tuple $\boldsymbol{R} = (R_1, \ldots, R_s) \in \mathbb{K}[X]^s$ of Lagrange interpolation polynomials, defined by the conditions

$$\deg(R_j) < n \quad \text{and} \quad R_j(x_i) = y_{i,j} \tag{1}$$

for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant s$, we can then combine Lemma 3 for all points so as to rewrite the vanishing condition of Problem 1 as a set of modular equations in $\mathbb{K}[X]$ as in Lemma 4 below. In what follows, we use the notation from Problem 1 and Theorem 1.

Lemma 4. *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 $\mathbb{Z}_{\geqslant 0}^s$ such that $|\boldsymbol{i}| < m$,*

$$Q^{[\boldsymbol{i}]}(X, \boldsymbol{R}) = 0 \mod \prod_{\substack{1 \leqslant r \leqslant n: \\ m_r > |\boldsymbol{i}|}} (X - x_r)^{m_r - |\boldsymbol{i}|}.$$

Proof. This result is easily obtained from Lemma 3 since the x_r are pairwise distinct. □

Note that when all multiplicities are equal, that is, $m = m_1 = \cdots = m_n$, for every $|\boldsymbol{i}|$ the modulus takes the simpler form $G^{m-|\boldsymbol{i}|}$, where $G = \prod_{1 \leqslant r \leqslant n}(X - x_r)$.

Writing $\boldsymbol{j} \cdot \boldsymbol{k} = j_1 k_1 + \cdots + j_s k_s$, recall from the statement of Theorem 1 that Γ is the set of all $\boldsymbol{j}$ in $\mathbb{Z}_{\geqslant 0}^s$ such that $|\boldsymbol{j}| \leqslant \ell$ and $\boldsymbol{j} \cdot \boldsymbol{k} < b$. Then, defining the positive integers

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

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 5. *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 $\boldsymbol{i} \in \mathbb{Z}_{\geqslant 0}^s$ 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

$$P_{\boldsymbol{i}} = \prod_{\substack{1 \leqslant r \leqslant n: \\ m_r > |\boldsymbol{i}|}} (X - x_r)^{m_r - |\boldsymbol{i}|} \tag{2a}$$

and

$$F_{\boldsymbol{i},\boldsymbol{j}} = \binom{\boldsymbol{j}}{\boldsymbol{i}} \boldsymbol{R}^{\boldsymbol{j}-\boldsymbol{i}} \bmod P_{\boldsymbol{i}}. \tag{2b}$$

It then follows from Lemmas 4 and 5 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,
- $\sum_{\boldsymbol{j} \in \Gamma} F_{\boldsymbol{i},\boldsymbol{j}} Q_{\boldsymbol{j}} = 0 \bmod P_{\boldsymbol{i}}$ for all $|\boldsymbol{i}| < m$.

Let now $M_{\boldsymbol{i}}$ be the positive integers given by

$$M_{\boldsymbol{i}} = \sum_{1 \leqslant r \leqslant n:\, m_r > |\boldsymbol{i}|} (m_r - |\boldsymbol{i}|),$$

for all $|\boldsymbol{i}| < m$. Since the $P_{\boldsymbol{i}}$ are monic polynomials of degree $M_{\boldsymbol{i}}$ 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

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

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

then choose arbitrary orders on the sets of indices $\{\boldsymbol{i} \in \mathbb{Z}_{\geqslant 0}^s \mid |\boldsymbol{i}| < m\}$ and Γ, that is, bijections

$$\phi : \{0, \ldots, \mu - 1\} \to \{\boldsymbol{i} \in \mathbb{Z}_{\geqslant 0}^s \mid |\boldsymbol{i}| < m\} \tag{3a}$$

and

$$\psi : \{0, \ldots, \nu - 1\} \to \Gamma; \tag{3b}$$

finally, for i in $\{0, \ldots, \mu - 1\}$ and j in $\{0, \ldots, \nu - 1\}$, associate $M'_i = M_{\phi(i)}$, $N'_j = N_{\psi(j)}$, $P'_i = P_{\phi(i)}$ and $F'_{i,j} = F_{\phi(i),\psi(j)}$. At this stage we have proved that the solutions to Problem 1 with input parameters $s, \ell, n, m_1, \ldots, m_n$, $b, k_1, \ldots, k_s$ and points $\{(x_i, y_{i,1}, \ldots, y_{i,s})\}_{1 \leqslant i \leqslant n}$ are exactly the solutions to Problem 2 with input parameters $\mu, \nu, M'_0, \ldots, M'_{\mu-1}, N'_0, \ldots, N'_{\nu-1}$ and polynomials $\{(P'_i, F'_{i,0}, \ldots, F'_{i,\nu-1})\}_{0 \leqslant i < \mu}$. This proves the correctness of Algorithm 1.

Proposition 6. *Algorithm 1 is correct and uses*

$$\mathcal{O}\big(\varrho \, \mathsf{M}(M) \log(M)\big)$$

operations in $\mathbb{K}$.

Proof. The only thing left to do is the complexity analysis; more precisely, giving an upper bound on the number of operations in $\mathbb{K}$ performed in Step **3**.

First, we need to compute $P_{\boldsymbol{i}}$ as in (2a) for every $\boldsymbol{i}$ in $\mathbb{Z}_{\geqslant 0}^s$ such that $|\boldsymbol{i}| < m$. This involves only m different polynomials $P_{\boldsymbol{i}_0}, \ldots, P_{\boldsymbol{i}_{m-1}}$ where we have chosen any indices $\boldsymbol{i}_j$ such that $|\boldsymbol{i}_j| = j$. We note that, defining for $j < m$ the polynomial $G_j = \prod_{1 \leqslant r \leqslant n:\, m_r > j} (X - x_r)$, we have $P_{\boldsymbol{i}_{m-1}} = G_{m-1}$ and for every $j < m - 1$, $P_{\boldsymbol{i}_j} = P_{\boldsymbol{i}_{j+1}} \cdot G_j$. The polynomials $G_0, \ldots, G_{m-1}$ have degree at most n and can be computed using $\mathcal{O}(m\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$; this is $\mathcal{O}(\varrho\mathsf{M}(M)\log(M))$ since $\varrho \geqslant \binom{s+m-1}{s} \geqslant m$ and $M = \sum_{1 \leqslant r \leqslant n} \binom{s+m_r}{s+1} \geqslant n$. Then $P_{\boldsymbol{i}_0}, \ldots, P_{\boldsymbol{i}_{m-1}}$ can be computed iteratively using $\mathcal{O}(\sum_{j<m} \mathsf{M}(\deg(P_{\boldsymbol{i}_j})))$ operations in $\mathbb{K}$; using the super-linearity of $\mathsf{M}(\cdot)$, this is $\mathcal{O}(\mathsf{M}(M))$ since $\deg(P_{\boldsymbol{i}_j}) = M_{\boldsymbol{i}_j}$ and $\sum_{j<m} M_{\boldsymbol{i}_j} \leqslant M$.

Then, we have to compute (some of) the interpolation polynomials $R_1, \ldots, R_s$. Due to Lemma 4, 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 $\mathcal{O}(|\Gamma|\mathsf{M}(n)\log(n)) \subseteq \mathcal{O}(\varrho\,\mathsf{M}(M)\log(M))$.

Finally, we compute $F_{\boldsymbol{i},\boldsymbol{j}}$ as in (2b) 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 2 with $\rho = \varrho$ and $M' = \sum_{0 \leqslant i < \mu} M'_i = M$. Noting that $M = \sum_{|\boldsymbol{i}|<m} M_{\boldsymbol{i}} = \sum_{1 \leqslant r \leqslant n} \binom{s+m_r}{s+1}$, we conclude that Theorem 2 implies Theorem 1.

III. Solving Problem 2 through structured linear systems

A. 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 [39].

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 and the anti-diagonal, respectively.

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 $M + N$. 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 $\Delta_{M,N}$).

A pair of 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 generator; 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

Algorithm 1. Reducing Problem 1 to Problem 2.

Input: $s, \ell, n, m_1, \ldots, m_n$ in $\mathbb{Z}_{>0}$, $b, k_1, \ldots, k_s$ in $\mathbb{Z}$, 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 pairwise distinct.

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

1. Compute $\Gamma = \{\boldsymbol{j} \in \mathbb{Z}^s_{\geqslant 0} \mid |\boldsymbol{j}| \leqslant \ell$ and $b - \boldsymbol{j} \cdot \boldsymbol{k} > 0\}$, $\mu = \binom{s+m-1}{s}$, $\nu = |\Gamma|$, and bijections ϕ and ψ as in (3)
2. Compute $M_{\boldsymbol{i}} = \sum_{1 \leqslant r \leqslant n:\, m_r > |\boldsymbol{i}|} (m_r - |\boldsymbol{i}|)$ and $N_{\boldsymbol{j}} = b - \boldsymbol{j} \cdot \boldsymbol{k}$ for $\boldsymbol{j} \in \Gamma$
3. Compute $P_{\boldsymbol{i}}$ and $F_{\boldsymbol{i},\boldsymbol{j}}$ for $|\boldsymbol{i}| < m$, $\boldsymbol{j} \in \Gamma$ as in (2)
4. Return the integers $\mu, \nu, M_{\phi(0)}, \ldots, M_{\phi(\mu-1)}, N_{\psi(0)}, \ldots, N_{\psi(\nu-1)}$ together with the polynomial tuples $\{(P_{\phi(i)}, F_{\phi(i),\psi(0)}, \ldots, F_{\phi(i),\psi(\nu-1)})\}_{0 \leqslant i < \mu}$

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 [8], 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 [7], Morf [35], Kaltofen [25], and Pan [39], 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 7. *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 [8, 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 [26], 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 [16], [55], [45], 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.

B. Solving Problem 2 through a mosaic-Hankel linear system

In this section, we give our first solution to Problem 2, thereby proving Theorem 2; this solution is outlined in Algorithm 2. It consists of first deriving and linearizing the modular equations of Lemma 8 below, and then solving the resulting mosaic-Hankel system using the approach recalled in Section III-A. Note that, when solving Problem 1 using the reduction to Problem 2 given in Section II, these modular equations are a generalization to arbitrary s of the extended key equations presented in [43], [54], [53] for $s = 1$.

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

(a) the Q_j are not all zero,
(b) for $0 \leqslant j < \nu$, $\deg(Q_j) < N'_j$,
(c) for $0 \leqslant i < \mu$, $\sum_{0 \leqslant j < \nu} F_{i,j} Q_j = 0 \bmod P_i$.

Our goal here is to linearize the condition *(c)* 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

$$N' \leqslant M' + 1. \tag{4}$$

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

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

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, $\delta_i > 0$ and $\gamma_j \geqslant 0$; 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 $0 \leqslant j < \nu$. (Those polynomials can be seen as a generalization of what is usually called *syndrome polynomials* in the context of coding theory; see for example [54].) 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*

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

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

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

For all i, j, the summand $F_{i,j} Q_j$ has degree less than $M'_i + N'_j - 1$, 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

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

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 [43], [54], we are going to rewrite the conditions in (5) 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 $\{\delta_i - M'_i, \ldots, \delta_i - 1\}$,

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

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

We will use the approach recalled in Section III-A 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 pair of matrices (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 7 and Lemma 9, this proves Theorem 2.

Lemma 10. *The displacement rank of A with respect to $\Delta'_{M',N'}$ is at most $\mu + \nu$. Furthermore, one can compute a corresponding generator of length $\mu + \nu$ for A using $\mathcal{O}((\mu+\nu)\mathsf{M}(M'))$ 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 $A - \mathcal{Z}_{M'} A \, \mathcal{Z}_{N'}$ 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 two integers i, K with $0 \leqslant i < K$, define $\mathcal{O}_{i,K} = [0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0]^T \in \mathbb{K}^K$ with 1 at position i, and

$$\mathcal{O}^{(V)} = [\mathcal{O}_{r_i, M'}]_{0 \leqslant i < \mu} \in \mathbb{K}^{M' \times \mu},$$
$$\mathcal{O}^{(W)} = \left[\mathcal{O}_{c_j, N'}\right]^T_{0 \leqslant j < \nu} \in \mathbb{K}^{\nu \times N'}.$$

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

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

and we define V' and W' as

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

Now, one can easily verify that the matrices

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

and

$$W = \begin{bmatrix} \mathcal{O}^{(W)} \\ W' \end{bmatrix} \in \mathbb{K}^{(\mu+\nu) \times N'} \tag{10b}$$

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

We notice that all we need in order 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 $0 \leqslant i < \mu$ and $0 \leqslant j < \nu$. 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

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

which is in $\mathcal{O}(\nu \mathsf{M}(M') + \mu \mathsf{M}(N'))$ using the super-linearity of M. Since we assumed in (4) that $N' \leqslant M' + 1$, this is $\mathcal{O}((\mu+\nu)\mathsf{M}(M'))$. □

Algorithm 2. Solving Problem 2 via a mosaic-Hankel linear system.

Input: positive integers $\mu,\ \nu,\ M'_0,\dots,M'_{\mu-1},\ N'_0,\dots,N'_{\nu-1}$ and polynomial tuples $\{(P_i,F_{i,0},\dots,F_{i,\nu-1})\}_{0\leqslant i<\mu}$ 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 $Q_0,\dots,Q_{\nu-1}$ in $\mathbb{K}[X]$ such that *(a)*, *(b)*, *(c)*.

1. For $i<\mu$, $j<\nu$, compute the coefficients $S_{i,j}^{(\gamma_j+r)}$ for $r<M'_i+N'_j-1$, that is, the coefficients of the polynomials $S^\star_{i,j}$ 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 7 with input V and W; if there is no solution then exit with no solution, otherwise find the coefficients of $Q_0,\dots,Q_{\nu-1}$
6. Return $Q_0,\dots,Q_{\nu-1}$

C. 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 $(P_i,F_{i,0},\dots,F_{i,\nu-1})_{0\leqslant i<\mu}$ and we look for polynomials $Q_0,\dots,Q_{\nu-1}$ in $\mathbb{K}[X]$ such that for $0\leqslant i<\mu$, $\sum_{0\leqslant j<\nu}F_{i,j}Q_j=0 \bmod P_i$, with the Q_j 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 $i<\mu$ 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 $h\in\mathbb{Z}_{\geqslant 0}$, let $\alpha_{i,j}^{(h)}\in\mathbb{K}^{M'_i}$ be the coefficient vector of the polynomial $X^hF_{i,j} \bmod P_i$, 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 $(Q_0,\dots,Q_{\nu-1})$ to Problem 2.*

Proof. By definition $A'_{i,j}$ is the $M'_i\times N'_j$ matrix of the mapping $Q\mapsto F_{i,j}Q \bmod P_i$, for Q in $\mathbb{K}[X]$ of degree less than N'_j. Thus, if $(Q_0,\dots,Q_{\nu-1})$ 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 $\sum_{0\leqslant j<\nu}F_{i,j}Q_j \bmod P_i$, for $i=0,\dots,\mu-1$. 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 7 and Lemma 11, this gives our second proof of Theorem 2.

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

Proof. We begin by giving two 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 $0\leqslant i<\mu$, we define

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

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

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 $0 \leqslant j < \nu$, let us further define

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

and

$$W'_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{K}^{N'}, \tag{13b}$$

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

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

Then, we claim that the matrices

$$Y = [\,-V \quad V'\,] \in \mathbb{K}^{M' \times (\mu+\nu)} \tag{14a}$$

and

$$Z = \begin{bmatrix} W^T A' \mathcal{Z}_{N'}^T \\ W'^T \end{bmatrix} \in \mathbb{K}^{(\mu+\nu) \times N'} \tag{14b}$$

are generators for A' for the Toeplitz-like displacement structure, that is,

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

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

$$B_{i,j} \;=\; \mathcal{C}_i A'_{i,j} \;=\; \left[\alpha_{i,j}^{(1)} \;\cdots\; \alpha_{i,j}^{(N'_j-1)} \;\alpha_{i,j}^{(N'_j)}\right] \in \mathbb{K}^{M'_i \times N'_j}.$$

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

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

To compute Y and Z, the only non-trivial steps are those giving V' and $W^T A'$. 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 $W^T A'$ 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 $X^h F_{i,j} \bmod P_i$ for $h = 0, \dots, N'_j - 1$ 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 $X^i F \bmod P$. Then, for $n \geqslant 1$ we can compute $c_0, \dots, c_{n-1}$ 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 $X^i F \bmod P = \sum_{0 \leqslant j < m} f_j \big(X^{i+j} \bmod P\big)$. 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 $c_0, \dots, c_{n-1}$ from $b_{m-1}, b_m, \dots, b_{m+n-2}$ 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 $b_m, \dots, b_{m+n-2}$. Writing P as $P = X^m + \sum_{0 \leqslant j < m} p_j X^j$ and using the fact that $X^i P \bmod P = 0$ for all $i \geqslant 0$, we see that the b_i 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, $b_m, \dots, b_{m+n-2}$ can be deduced from $b_0, \dots, b_{m-1}$ 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 [46, Theorem 3.1]. □

IV. Applications to the decoding of Reed-Solomon codes

To conclude, we discuss Theorem 1 in specific contexts related to the decoding of Reed-Solomon codes; in this section

Algorithm 3. Solving Problem 2 via a Toeplitz-like linear system.

Input: positive integers $\mu,\ \nu,\ M'_0,\dots,M'_{\mu-1},\ N'_0,\dots,N'_{\nu-1}$ and polynomial tuples $\{(P_i,F_{i,0},\dots,F_{i,\nu-1})\}_{0\leqslant i<\mu}$ 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 $Q_0,\dots,Q_{\nu-1}$ 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 $i<\mu$, $j<\nu-1$
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 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 7 with input Y and Z; if there is no solution then exit with no solution, otherwise find the coefficients of $Q_0,\dots,Q_{\nu-1}$
9. Return $Q_0,\dots,Q_{\nu-1}$

we always have $s=1$. First, we give our complexity result in the case of list-decoding via the Guruswami-Sudan algorithm [23]; then we show how the re-encoding technique [29], [27] can be used in our setting; then, we discuss the interpolation step of the Wu algorithm [52]; and finally we present the application of our results to the interpolation step of the soft-decoding [28]. In these contexts of applications, we will use some of the assumptions on the parameters $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$ given in Section I. Note that in the context of soft-decoding, the x_i in the input of Problem 1 are not necessarily pairwise distinct: we will explain how to adapt our algorithms to this case. Besides, still in this context, the number of points n is no longer equal to the length of the code and may actually be much larger, unlike in hard-decision (list-)decoding.

A. Interpolation step of the Guruswami-Sudan algorithm

We study here the specific context of the interpolation step of the Guruswami-Sudan list-decoding algorithm for Reed-Solomon codes. This interpolation step is precisely Problem 1 where we have $s=1$ and we make assumptions $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$. Under $\mathbf{H_2}$, the set Γ introduced in Theorem 1 reduces to $\{j\in\mathbb{Z}_{\geqslant 0}\ :\ j\leqslant\ell\}=\{0,\dots,\ell\}$, so that $|\Gamma|=\ell+1$. Thus, assumption $\mathbf{H_1}$ ensures that the parameter ϱ in that theorem is $\varrho=\ell+1$; because of $\mathbf{H_4}$ all multiplicities are equal so that we further have $M=\binom{m+1}{2}n=\frac{m(m+1)}{2}n$. From Theorem 1, we obtain the following result, which substantiates our claimed cost bound in Section I, Table I.

Corollary 14. *Taking $s=1$, if the parameters $\ell,n,\ m:=m_1=\cdots=m_n,b$ and $k:=k_1$ satisfy $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$, then there exists a probabilistic algorithm that computes a solution to Problem 1 using*

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

operations in $\mathbb{K}$, with probability of success at least $1/2$.

We note that the probability analysis in Theorem 1 is simplified in this context. Indeed, to ensure probability of success at least $1/2$, the algorithm chooses $\mathcal{O}(m^2n)$ elements uniformly at random in a set $S\subseteq\mathbb{K}$ of cardinality at least $24m^4n^2$; if $|\mathbb{K}|<24m^4n^2$, one can use the remarks following Theorem 1 in Section I about solving the problem over an extension of $\mathbb{K}$ and retrieving a solution over $\mathbb{K}$. Here, the base field $\mathbb{K}$ of a Reed-Solomon code must be of cardinality at least n since the x_i are distinct; then, an extension degree $d=\mathcal{O}(\log_n(m))$ suffices and the cost bound above becomes $\mathcal{O}\big(\ell^{\omega-1}\mathsf{M}(m^2n)\log(mn)^2\cdot\mathsf{M}(d)\log(d)\big)$. Besides, in the list-decoding of Reed-Solomon codes we have $m=\mathcal{O}(n^2)$, so that $d=\mathcal{O}(1)$ and the cost bound and probability of success in Corollary 14 hold for *any* field $\mathbb{K}$ (of cardinality at least n).

B. Re-encoding technique

The re-encoding technique has been introduced by Koetter and Vardy [29], [27] in order to reduce the cost of the interpolation step in list- and soft-decoding of Reed-Solomon codes. Here, for the sake of clarity, we present this technique only in the context of Reed-Solomon list-decoding via the Guruswami-Sudan algorithm, using the same notation and assumptions as in Subsection IV-A above: $s=1$ and we have $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$. Under some additional assumption on the input points in Problem 1, by means of partially pre-solving the problem one obtains an interpolation problem whose linearization has smaller dimensions. The idea at the core of this technique is summarized in the following lemma [29, Lemma 4].

Lemma 15. *Let m be a positive integer, x be an element in $\mathbb{K}$, and $Q=\sum_j Q_j(X)Y^j$ be a polynomial in $\mathbb{K}[X,Y]$. Then, $Q(x,0)=0$ with multiplicity at least m if and only if $(X-x)^{m-j}$ divides Q_j for each $j<m$.*

Proof. By definition, $Q(x,0)=0$ with multiplicity at least m if and only if $Q(X+x,Y)$ has no monomial of total degree less than m. Since $Q(X+x,Y)=\sum_j Q_j(X+x)Y^j$, this is equivalent to the fact that X^{m-j} divides $Q_j(X+x)$ for each $j<m$. $\square$

This property can be generalized to the case of several roots of the form $(x, 0)$. More precisely, the re-encoding technique is based on a shift of the received word by a well-chosen code word, which allows us to ensure the following assumption on the points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n}$: for some integer $n_0 \geqslant k+1$,

$$y_1 = \cdots = y_{n_0} = 0 \quad \text{and} \quad y_{n_0+1} \neq 0, \ldots, y_n \neq 0. \qquad (15)$$

We now define the polynomial $G_0 = \prod_{1\leqslant r\leqslant n_0}(X - x_r)$ which vanishes at x_r when $y_r = 0$, and Lemma 15 can be rewritten as follows: $Q(x_r, 0) = 0$ with multiplicity at least m for $1 \leqslant r \leqslant n_0$ if and only if G_0^{m-j} divides Q_j for each $j < m$. Thus, we know how to solve the vanishing condition for the n_0 points for which $y_r = 0$: by setting each of the m polynomials $Q_0, \ldots, Q_{m-1}$ as the product of a power of G_0 and an unknown polynomial. Combining this with the polynomial approximation problem corresponding to the points $\{(x_r, y_r)\}_{n_0+1\leqslant r\leqslant n}$, there remains to solve a smaller approximation problem.

Indeed, under the previously mentioned assumptions $s = 1$ and $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$, it has been shown in Section II that the vanishing condition *(iv)* of Problem 1 restricted to points $\{(x_r, y_r)\}_{n_0+1\leqslant r\leqslant n}$ is equivalent to the simultaneous polynomial approximations

$$\sum_{i\leqslant j\leqslant \ell} \binom{j}{i} R^{j-i} Q_j = 0 \mod G^{m-i} \qquad \text{for } i < m,$$

where $G = \prod_{n_0+1\leqslant r\leqslant n}(X - x_r)$ and R is the interpolation polynomial such that $\deg R < n - n_0$ and $R(x_r) = y_r$ for $n_0 + 1 \leqslant r \leqslant n$. On the other hand, we have seen that the vanishing condition for the points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n_0}$ is equivalent to $Q_j = G_0^{m-j} Q_j^\star$ for each $j < m$, for some unknown polynomials $Q_0^\star, \ldots, Q_{m-1}^\star$. Combining both equivalences, we obtain for $i < m$

$$\sum_{i\leqslant j<m} F_{i,j}\, Q_j^\star + \sum_{m\leqslant j\leqslant \ell} F_{i,j}\, Q_j = 0 \mod G^{m-i} \qquad (16)$$

with

$$F_{i,j} = \begin{cases} \binom{j}{i} R^{j-i} G_0^{m-j} \bmod G^{m-i} & \text{for } i \leqslant j < m, \\ \binom{j}{i} R^{j-i} \bmod G^{m-i} & \text{for } m \leqslant j \leqslant \ell. \end{cases} \qquad (17)$$

Obviously, the degree constraints on $Q_0, \ldots, Q_{m-1}$ directly correspond to degree constraints on $Q_0^\star, \ldots, Q_{m-1}^\star$ while those on $Q_m, \ldots, Q_\ell$ are unchanged. The number of equations obtained when linearizing (16) is $M' = \sum_{i<m} \deg(G^{m-i}) = \frac{m(m+1)}{2}(n - n_0)$, while the number of unknowns is $N' = \sum_{j<m}(b - jk - (m-j)n_0) + \sum_{m\leqslant j\leqslant \ell}(b - jk) = \sum_{j\leqslant \ell}(b - jk) - \frac{m(m+1)}{2} n_0$. In other words, we have reduced the number of (linear) unknowns as well as the number of (linear) equations by the same quantity $\frac{m(m+1)}{2} n_0$, which is the number of linear equations used to express the vanishing condition for the n_0 points $(x_1, 0), \ldots, (x_{n_0}, 0)$. (Note that if we were in the more general context of possibly distinct multiplicities, we would have set $y_i = 0$ for the n_0 points which have the highest multiplicities, in order to maximize the benefit of the re-encoding technique.)

This re-encoding technique is summarized in Algorithm 4. Assuming that Step **4** is done using Algorithm 2 or 3, we obtain the following result about list-decoding of Reed-Solomon codes using the re-encoding technique.

Corollary 16. *Take $s = 1$ and assume the parameters ℓ, n, $m := m_1 = \cdots = m_n, b$ and $k := k_1$ satisfy $\boldsymbol{H_1}$, $\boldsymbol{H_2}$, $\boldsymbol{H_3}$, $\boldsymbol{H_4}$. Assume further that the points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n}$ satisfy* (15) *for some $n_0 \geqslant k+1$. Then there exists a probabilistic algorithm that computes a solution to Problem 1 using*

$$\mathcal{O}\big(\ell^{\omega-1}\mathsf{M}(m^2(n-n_0))\log(n-n_0)^2 + m\mathsf{M}(mn_0) + \mathsf{M}(n_0)\log(n_0)\big) \subseteq \mathcal{O}\tilde{~}(\ell^{\omega-1}m^2(n-n_0) + m^2 n_0)$$

operations in $\mathbb{K}$ with probability of success at least $1/2$.

Proof. For Steps **1**, **2**, **3**, the complexity analysis is similar to the one in the proof of Proposition 6; we still note that we have to compute G_0, so that these steps use $\mathcal{O}(\ell\mathsf{M}(m^2(n-n_0))\log(n-n_0) + \mathsf{M}(n_0)\log(n_0))$ operations in $\mathbb{K}$. According to Theorem 2, Step **4** uses $\mathcal{O}\big(\ell^{\omega-1}\mathsf{M}(m^2(n-n_0))\log(n-n_0)^2\big)$ operations in $\mathbb{K}$. Step **5** uses $\mathcal{O}(m\mathsf{M}(mn_0) + \mathsf{M}(m^2(n-n_0)))$ operations in $\mathbb{K}$. Indeed, we first compute $G_0, \ldots, G_0^m$ using $\mathcal{O}(m\mathsf{M}(mn_0))$ operations and then the products $G_0^{m-j}Q_j$ for $j < m$ are computed using $\mathcal{O}(m\mathsf{M}(mn_0) + \mathsf{M}(m^2(n-n_0)))$ operations: for each $j < m$, the product $G_0^{m-j}Q_j$ can be computed using $\mathcal{O}(\mathsf{M}(mn_0) + \mathsf{M}(\deg(Q_j)))$ operations since G_0^{m-j} has degree at most mn_0; and from Algorithms 2 and 3 we know that $\deg Q_0 + \cdots + \deg Q_{m-1} \leqslant (\sum_{i<m} M_i') + 1$ (see (4) in Section III-B), with here $\sum_{i<m} M_i' = \frac{m(m+1)}{2}(n - n_0)$. ☐

Similarly to the remarks following Corollary 14, if $|\mathbb{K}| < 24m^2(n - n_0)$ then $\mathbb{K}$ does not contain enough elements to ensure a probability of success at least $1/2$ using our algorithms, but one can solve the problem over an extension of degree $\mathcal{O}(1)$ and retrieve a solution over $\mathbb{K}$ without impacting the cost bound.

C. Interpolation step in the Wu algorithm

Our goal now is to show that our algorithms can also be used to efficiently solve the interpolation step in the Wu algorithm. In this context, we have $s = 1$ and we make assumptions $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_4}$ on input parameters to Problem 1. We note that here the weight k is no longer related to the dimension of the code; besides, we may have $k \leqslant 0$.

Roughly, the Wu algorithm [52] works as follows. It first uses the Berlekamp-Massey algorithm to reduce the problem of list-decoding a Reed-Solomon code to a problem of rational reconstruction which focuses on the error locations (while the Guruswami-Sudan algorithm directly relies on a problem of polynomial reconstruction which focuses on the correct locations). Then, it solves this problem using an interpolation step and a root-finding step which are very similar to the ones in the Guruswami-Sudan algorithm.

Here we focus on the interpolation step, which differs from the one in the Guruswami-Sudan algorithm by mainly one feature: the points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n}$ lie in $\mathbb{K} \times (\mathbb{K} \cup \{\infty\})$, that is, some y_r may take the special value ∞. For a point

Algorithm 4. Interpolation step of list-decoding Reed-Solomon codes using re-encoding.

Input: ℓ, n, m, b, k in $\mathbb{Z}_{>0}$ and satisfying $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$, and points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n}$ in $\mathbb{K}^2$ with the x_r pairwise distinct and the y_r satisfying (15).

Output: $Q_0, \ldots, Q_\ell$ in $\mathbb{K}[X]$ such that $\sum_{j\leqslant \ell} Q_j Y^j$ is a solution to Problem 1 with input $s = 1$, $\ell, n, m = m_1 = \cdots = m_n, b, k$ and $\{(x_r, y_r)\}_{1\leqslant r\leqslant n}$.

1. Compute $\mu = m, \nu = \ell + 1, M'_i = (m - i)(n - n_0)$, $N'_j = b - jk - n_0(m - j)$ for $j < m$ and $N'_j = b - jk$ for $m \leqslant j \leqslant \ell$
2. Compute $G_0 = \prod_{1\leqslant r\leqslant n_0}(X - x_r)$ and $P_i = \left(\prod_{n_0+1\leqslant r\leqslant n}(X - x_r)\right)^{m-i}$ for $i < m$
3. Compute the $F_{i,j}$ for $i < m$ and $j \leqslant \ell$ as in (17)
4. Compute a solution $Q_0, \ldots, Q_\ell$ to Problem 2 on input μ, ν, $M'_0, \ldots, M'_{m-1}, N'_0, \ldots, N'_\ell$ and the polynomials $\{(P_i, F_{i,0}, \ldots, F_{i,\ell})\}_{0\leqslant i<m}$
5. Return $G_0^m Q_0, G_0^{m-1} Q_1, \ldots, G_0 Q_{m-1}, Q_m, \ldots, Q_\ell$, or report "no solution" if Step **4** did

(x, ∞), a polynomial Q in $\mathbb{K}[X, Y]$ and a parameter ℓ such that $\deg_Y(Q) \leqslant \ell$, Wu defines in [52] the vanishing condition $Q(x, \infty) = 0$ with multiplicity at least m as the vanishing condition $\overline{Q}(x, 0) = 0$ with multiplicity at least m, where $\overline{Q} = Y^\ell Q(X, Y^{-1})$ is the reversal of Q with respect to the variable Y and the parameter ℓ. Thus, we have the following direct adaptation of Lemma 15.

Lemma 17. *Let ℓ, m be positive integers, x be an element in $\mathbb{K}$, and $Q = \sum_{j\leqslant \ell} Q_j(X) Y^j$ be a polynomial in $\mathbb{K}[X, Y]$ with $\deg_Y(Q) \leqslant \ell$. Then, $Q(x, \infty) = 0$ with multiplicity at least m if and only if $(X - x)^{m-j}$ divides $Q_{\ell-j}$ for each $j < m$.*

As in the re-encoding technique, assuming we reorder the points so that $y_1 = \cdots = y_{n_\infty} = \infty$ and $y_r \neq \infty$ for $r > n_\infty$ for some $n_\infty \geqslant 0$, the vanishing condition of Problem 1 restricted to the points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n_\infty}$ is equivalent to $Q_{\ell-j} = G_\infty^{m-j} Q^\star_{\ell-j}$ for each $j < m$, for some unknown polynomials $Q^\star_{\ell-m+1}, \ldots, Q^\star_\ell$. The degree constraints on $Q_{\ell-m+1}, \ldots, Q_\ell$ directly correspond to degree constraints on $Q^\star_{\ell-m+1}, \ldots, Q^\star_\ell$, while those of $Q_0, \ldots, Q_{\ell-m}$ are unchanged.

This means that in the interpolation problem we are faced with, we can deal with the points of the form (x, ∞) the same way we dealt with the points of the form $(x, 0)$ in the case of the re-encoding technique: we can pre-solve the corresponding equations efficiently, and we are left with an approximation problem whose dimensions are smaller than if no special attention had been paid when dealing with the points of the form (x, ∞). More precisely, let $G_\infty = \prod_{1\leqslant r\leqslant n_\infty}(X - x_r)$ as well as $G = \prod_{n_\infty+1\leqslant r\leqslant n}(X - x_r)$ and R of degree less than $n - n_\infty$ such that $R(x_r) = y_r$ for each $r > n_\infty$. Defining further

$$F_{i,j} = \begin{cases} \binom{j}{i} R^{j-i} & \text{for } i \leqslant j \leqslant \ell - m, \\ \binom{j}{i} R^{j-i} G_\infty^{j-\ell+m} & \text{for } \ell - m < j \leqslant \ell, \end{cases}$$

we obtain the following simultaneous polynomial approximations: for $i < m$,

$$\sum_{i\leqslant j\leqslant \ell-m} F_{i,j}\, Q_j + \sum_{\ell-m<j\leqslant \ell} F_{i,j}\, Q^\star_j = 0 \mod G^{m-i}.$$

Pre-solving the equations for the points of the form (x, ∞) has led to reduce the number of (linear) unknowns as well as the number of (linear) equations by the same quantity $\frac{m(m+1)}{2} n_\infty$, which is the number of linear equations used to express the vanishing condition for the n_∞ points $(x_1, \infty), \ldots, (x_{n_\infty}, \infty)$. We have the following result.

Corollary 18. *Take $s = 1$ and assume that the parameters ℓ, n, $m := m_1 = \cdots = m_n, b$ and $k := k_1$ satisfy $\boldsymbol{H_1}$, $\boldsymbol{H_2}$, $\boldsymbol{H_4}$. Assume further that each of the points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n}$ is allowed to have the special value $y_r = \infty$. Then there exists a probabilistic algorithm that computes a solution to Problem 1 using*

$$\mathcal{O}\big(\ell^{\omega-1} \mathsf{M}(m^2 n) \log(n)^2\big) \subseteq \mathcal{O}\tilde{~}(\ell^{\omega-1} m^2 n)$$

operations in $\mathbb{K}$ with probability of success is at least $1/2$.

As above, if $|\mathbb{K}| < 24m^2(n - n_\infty)$ then in order to ensure a probability of success at least $1/2$ using our algorithms, one can solve the problem over an extension of degree $\mathcal{O}(1)$ and retrieve a solution over $\mathbb{K}$, without impacting the cost bound.

We note that unlike in the re-encoding technique where the focus was on a reduced cost involving $n - n_0$, here we are not interested in writing the detailed cost involving $n - n_\infty$. The reason is that n_∞ is expected to be close to 0 in practice. The main advantage of the Wu algorithm over the Guruswami-Sudan algorithm is that it uses a smaller multiplicity m, at least for practical code parameters; details about the choice of parameters m and ℓ in the context of the Wu algorithm can be found in [5, Section IV.C].

D. Application to soft-decoding of Reed-Solomon codes

As a last application, we briefly sketch how to adapt our results to the context of soft-decoding, in which we still have $s = 1$. The interpolation step in soft-decoding of Reed-Solomon codes [28] differs from Problem 1 because there is no assumption ensuring that the x_r are pairwise distinct among the points $\{(x_r, y_r)\}_{1\leqslant r\leqslant n}$. Regarding our algorithms, this is not a minor issue since this assumption is at the core of the reduction in Section II; we will see that we can still rely on Problem 2 in this context. However, although the number of linear equations $\sum_{1\leqslant r\leqslant n}\frac{m_r(m_r+1)}{2}$ imposed by the vanishing condition is not changed by the fact that several x_r can be the same field element, it is expected that the reduction to Problem 2 will not be as effective as before. More precisely, the displacement rank of the structured matrix in the linearizations of the problem in Algorithms 2 and 3 may in some cases be larger than if the x_r were pairwise distinct.

To measure to which extent we are far from the situation where the x_r are pairwise distinct, we use the parameter

$$q = \max_{x\in\mathbb{K}} \left|\{r \in \{1,\ldots,n\} \;\mid\; x_r = x\}\right| .$$

For example, $q = 1$ corresponds to pairwise distinct x_r, while $q = n$ corresponds to $x_1 = \cdots = x_n$; we always have $q \leqslant n$ and, if $\mathbb{K}$ is a finite field, $q \leqslant |\mathbb{K}|^s$ with $s = 1$ in our context here. Then, we can write the set of points $\mathcal{P} = \{(x_r, y_r)\}_{1\leqslant r\leqslant n}$ as the disjoint union of q sets $\mathcal{P} = \mathcal{P}_1 \cup\cdots\cup \mathcal{P}_q$ where each set $\mathcal{P}_h = \{(x_{h,r}, y_{h,r})\}_{1\leqslant r\leqslant n_h}$ is such that the $x_{h,r}$ are pairwise distinct; we denote $m_{h,r}$ the multiplicity associated with the point $(x_{h,r}, y_{h,r})$ in the input of Problem 1. Now, the vanishing condition *(iv)* asks that the q vanishing conditions restricted to each $\mathcal{P}_h$ hold simultaneously. Indeed, $Q(x_r, y_r) = 0$ with multiplicity at least m_r for all points (x_r, y_r) in $\mathcal{P}$ if and only if for each set $\mathcal{P}_h$, $Q(x_{h,r}, y_{h,r}) = 0$ with multiplicity at least $m_{h,r}$ for all points $(x_{h,r}, y_{h,r})$ in $\mathcal{P}_h$.

We have seen in Section II how to rewrite the vanishing condition as simultaneous polynomial approximations when the x_r are pairwise distinct. This reduction extends to this case: by simultaneously rewriting the vanishing condition for each set $\mathcal{P}_h$, one obtains a problem of simultaneous polynomial approximations whose solutions exactly correspond to the solutions of the instance of (extended) Problem 1 we are considering. Here, we do not give details about this reduction; they can be found in [53, Section 5.1.1]. Now, let $m^{(h)}$ be the largest multiplicity among those of the points in $\mathcal{P}_h$; in this reduction to Problem 2, the number of polynomial equations we obtain is $\sum_{1\leqslant h\leqslant q} m^{(h)}$. Thus, according to Theorem 2, for solving this instance of Problem 2, our Algorithms 2 and 3 use $\mathcal{O}\tilde{~}(\rho^{\omega-1}M')$ operations in $\mathbb{K}$, where $\rho = \max(\ell+1, \sum_{1\leqslant h\leqslant q} m^{(h)})$ and $M' = \sum_{1\leqslant r\leqslant n}\frac{m_r(m_r+1)}{2}$. We see in this cost bound that the distribution of the points into disjoint sets $\mathcal{P} = \mathcal{P}_1 \cup\cdots\cup \mathcal{P}_q$ has an impact on the number of polynomial equations in the instance of Problem 2 we get: when choosing this distribution, multiplicities could be taken into account in order to minimize this impact.

APPENDIX A
ON ASSUMPTION $\mathbf{H_1}$

In this appendix, we discuss the relevance of the assumption $\mathbf{H_1}$ introduced previously for Problem 1. In the introduction, we did not make any assumption on $m = \max_{1\leqslant i\leqslant n} m_i$ and ℓ, but we mentioned that the assumption $\mathbf{H_1}$, that is, $m \leqslant \ell$ is mostly harmless. The following lemma substantiates this claim, by showing that the case $m > \ell$ can be reduced to the case $m = \ell$.

Lemma 19. *Let $s, \ell, n, m_1, \ldots, m_n, b, \boldsymbol{k}$ be parameters for Problem 1, and suppose that $m > \ell$. Define $P = \prod_{1\leqslant i\leqslant n:\, m_i>\ell}(X-x_i)^{m_i-\ell}$ and $d = \deg(P)$. The solutions to this problem are the polynomials of the form $Q = Q^\star P$ with $Q^\star$ a solution for the parameters $s, \ell, n, m'_1, \ldots, m'_n, b-d, \boldsymbol{k}$, where $m'_i = \ell$ if $m_i > \ell$ and $m'_i = m_i$ otherwise.*

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 $h \geqslant m_i - \ell$, since $|\boldsymbol{j}| \leqslant \ell$ by condition *(ii)* and $h + |\boldsymbol{j}| \geqslant m_i$ by condition *(iv)*. Therefore, if $m_i > \ell$ then $X^{m_i-\ell}$ divides Q_i and, shifting back the coordinates for each i, we deduce that P divides Q.

Let us now consider the polynomial $Q^\star = Q/P$ and show that it solves Problem 1 for the parameters $s, \ell, n, m'_1, \ldots, m'_n, b-d, \boldsymbol{k}$. First, $Q^\star$ clearly satisfies conditions *(i)* and *(ii)*. Furthermore, writing $Q = \sum_{\boldsymbol{j}} Q_{\boldsymbol{j}}(X)\boldsymbol{Y}^{\boldsymbol{j}}$ and $Q^\star = \sum_{\boldsymbol{j}} Q^\star_{\boldsymbol{j}}(X)\boldsymbol{Y}^{\boldsymbol{j}}$, we have $Q^\star_{\boldsymbol{j}} = Q_{\boldsymbol{j}}/P$ for all $\boldsymbol{j}$, so that

$$\begin{aligned}\mathrm{wdeg}_{\boldsymbol{k}}(Q^\star) &= \max_{\boldsymbol{j}}(\deg(Q_{\boldsymbol{j}}) - d + k_1 j_1 + \cdots + k_s j_s)\\ &= \mathrm{wdeg}_{\boldsymbol{k}}(Q) - d\\ &< b-d,\end{aligned}$$

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

$$Q^\star_i(X,\boldsymbol{Y}) = \frac{Q_i(X,\boldsymbol{Y})}{X^{m_i-\ell}\,P_i(X)},$$

where

$$P_i(X) = \prod_{h\neq i:\ m_h>\ell}(X + x_i - x_h)^{m_h-\ell}.$$

All the monomials of $Q_i(X,\boldsymbol{Y})/X^{m_i-\ell}$ have the form $X^h\boldsymbol{Y}^{\boldsymbol{j}}$ with $h + |\boldsymbol{j}| \geqslant m_i - (m_i - \ell) = \ell$ and, since $P_i(0) \neq 0$, the same holds for $Q^\star_i(X,\boldsymbol{Y})$.

Conversely, let Q' be *any* solution to Problem 1 with parameters $s, \ell, n, m'_1, \ldots, m'_n, b-d, \boldsymbol{k}$. Proceeding as in the previous paragraph, one easily verifies that the product $Q'P$ is a solution to Problem 1 with parameters $s, \ell, n, m_1, \ldots, m_n, b, \boldsymbol{k}$. □

APPENDIX B
ON ASSUMPTION $\mathbf{H_3}$

In this appendix, we show the relevance of the assumption "$k_j < n$ for some $j \in \{1,\ldots,s\}$" when considering Problem 1; in particular when $s = 1$ or when we assume that $k_1 = \cdots = k_s =: k$, this shows the relevance of the assumption $\mathbf{H_3}$: $0 \leqslant k < n$. More precisely, when $k_j \geqslant n$ for every j, Lemma 20 below gives an explicit solution to Problem 1.

Lemma 20. *Let $s, \ell, n, \boldsymbol{m}, b, \boldsymbol{k}$ be parameters for Problem 1 and suppose that $k_j \geqslant n$ for $j = 1,\ldots,s$. Define $P =$*

$\prod_{1\leqslant i\leqslant n}(X-x_i)^{m_i}$ and $d=\deg(P)=\sum_{1\leqslant i\leqslant n} m_i$. If $b\leqslant d$ then this problem has no solution. Otherwise, a solution is given by the polynomial P (considered as an element of $\mathbb{K}[X,\boldsymbol{Y}]$).

Proof. If $b>d$ then it is easily checked that P 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 $b>d$ must hold. Let $d_Y=\deg_{\boldsymbol{Y}} Q$. If $d_Y\geqslant m=\max_i m_i$, then the weighted-degree condition *(iii)* gives $b>\operatorname{wdeg}_{\boldsymbol{k}}(Q)\geqslant d_Y(\min_j k_j)\geqslant mn\geqslant d$. Let us finally assume $d_Y<m$. Following the proof of Lemma 19, we can write $Q=P^\star Q^\star$ where $P^\star=\prod_{1\leqslant i\leqslant n:\, m_i>d_Y}(X-x_i)^{m_i-d_Y}$, for some $Q^\star$ in $\mathbb{K}[X,\boldsymbol{Y}]$ such that $\deg_{\boldsymbol{Y}} Q^\star=d_Y$. Then, the weighted-degree condition gives $b>\sum_{1\leqslant i\leqslant n:\, m_i>d_Y}(m_i-d_Y)+\operatorname{wdeg}_{\boldsymbol{k}}(Q^\star)\geqslant\sum_{1\leqslant i\leqslant n:\, m_i>d_Y}(m_i-d_Y)+d_Y n\geqslant\sum_{1\leqslant i\leqslant n:\, m_i>d_Y} m_i+\sum_{1\leqslant i\leqslant n:\, m_i\leqslant d_Y} d_Y\geqslant d$. □

Appendix C
The lattice-based approach

In this appendix, we summarize the approach for solving Problem 1 via the computation of a reduced polynomial lattice basis; this helps us to compare the cost bounds for this approach with the cost bound we give in Theorem 1. Here, $s\geqslant 1$ and for simplicity, we assume that $k:=k_1=\cdots=k_s$ as in the list-decoding of folded Reed-Solomon codes. Besides, we make the assumptions $\mathbf{H_1}$, $\mathbf{H_2}$, $\mathbf{H_3}$, $\mathbf{H_4}$ as presented in the introduction. Two main lattice constructions exist in the literature; following [10, §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.

Let $G=\prod_{1\leqslant r\leqslant n}(X-x_r)$ and $R_1,\ldots,R_s\in\mathbb{K}[X]$ such that $\deg(R_j)<n$ and $R_j(x_i)=y_{i,j}$ for every $j\in\{1,\ldots,s\}$ and $i\in\{1,\ldots,n\}$. In the first construction, the lattice is generated by the polynomials

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

this construction may be called *banded* due to the shape of the generators above when $s=1$. In the second construction, which may be called *triangular*, the lattice is generated by the polynomials

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

When $s=1$, the first construction is used in [4, Remark 16] and [32], [13], and the second one is used in [4], [6]; when $s\geqslant 1$, the former can be found in [10] while the latter appears in [9], [14]. In both cases the actual lattice bases are the coefficient vectors (in $\boldsymbol{Y}$) of the polynomials $h(X,X^kY_1,\ldots,X^kY_s)$, for h in either of the sets above; these X^k are introduced to account for the weighted-degree condition (iii) in Problem 1.

In this context, for a lattice of dimension L given by generators of degree at most d, the algorithm in [20] 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, see the algorithm of Gupta, Sarkar, Storjohann, and Valeriote [21], whose cost is $\mathcal{O}(L^\omega\mathsf{M}(d)(\log(L)^2+\log(d)))$.

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

$$L_B=\binom{s+m-1}{s}+\binom{s+m-1}{s-1}\binom{s+\ell-m}{s}$$

and

$$d_B=\mathcal{O}(mn).$$

The dimension formula is given explicitly in [10, 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 [20] 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 [6, 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 $i_1,i_2\in\mathbb{Z}_{\geqslant 0}$. 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 $n+2$ 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 [19, §14.9]). Then, with the algorithm of [20] 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 [44, 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 [20] 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 $s = 1$, the assumption $\mathbf{H_1}$ implies that these costs are $\mathcal{O}(\ell^\omega \mathsf{M}(mn)\log(\ell n))$ and $\mathcal{O}(\ell^\omega \mathsf{M}(\ell n)\log(\ell n))$, respectively, as reported in [13], [6]. For $s > 1$, the costs obtained in [10], [9] are worse, but only because the short vector algorithms used in those references are slower than the ones we refer to; no cost bound is explicitly given in [14]. The result in Theorem 1 is an improvement over those of both [10] and [9]. 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 [10], [9] are at least $\binom{s+\ell}{s}^{\omega} mn$; a simplification proves our claim.

Acknowledgment

M. F. I. Chowdhury and É. Schost were supported by NSERC and by the Canada Research Chairs program. V. Neiger was supported by an international mobility grant Explo'ra Doc from Région Rhône-Alpes. C.-P. Jeannerod and G. Villard were partly supported by the ANR HPAC project (ANR 11 BS02 013).

We thank the two reviewers for their thorough reading and helpful comments. We also thank the three reviewers of the preliminary version [12] of this work, 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 Trans. Inf. Theory*, vol. 51, no. 7, pp. 2257–2265, Jul. 2005.

[2] B. Beckermann, "A reliable method for computing M-Padé approximants on arbitrary staircases," *J. Comput. Appl. Math.*, vol. 40, no. 1, pp. 19–42, 1992.

[3] B. Beckermann and G. Labahn, "A uniform approach for the fast computation of matrix-type Padé approximants," *SIAM J. Matrix Anal. Appl.*, vol. 15, no. 3, pp. 804–823, Jul. 1994. [Online]. Available: http://dx.doi.org/10.1137/S0895479892230031

[4] P. Beelen and K. Brander, "Key equations for list decoding of Reed-Solomon codes and how to solve them," *J. Symbolic Comput.*, vol. 45, no. 7, pp. 773–786, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747717110000477

[5] P. Beelen, T. Høholdt, J. S. R. Nielsen, and Y. Wu, "On rational interpolation-based list-decoding and list-decoding binary Goppa codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3269–3281, 2013.

[6] D. J. Bernstein, "Simplified high-speed high-distance list decoding for alternant codes," in *PQCrypto'11*, ser. LNCS, vol. 7071. Springer, 2011, pp. 200–216.

[7] R. R. Bitmead and B. D. O. Anderson, "Asymptotically fast solution of Toeplitz and related systems of linear equations," *Linear Algebra Appl.*, vol. 34, pp. 103–116, 1980.

[8] A. Bostan, C.-P. Jeannerod, and E. Schost, "Solving structured linear systems with large displacement rank," *Theor. Comput. Sci.*, vol. 407, no. 1-3, pp. 155–181, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2008.05.014

[9] K. Brander, "Interpolation and list decoding of algebraic codes," Ph.D. dissertation, Technical University of Denmark, 2010.

[10] P. Busse, "Multivariate list decoding of evaluation codes with a Gröbner basis perspective," Ph.D. dissertation, University of Kentucky, 2008.

[11] D. G. Cantor and E. Kaltofen, "On fast multiplication of polynomials over arbitrary algebras," *Acta Inform.*, vol. 28, no. 7, pp. 693–701, 1991. [Online]. Available: http://dx.doi.org/10.1007/BF01178683

[12] M. F. I. Chowdhury, C.-P. Jeannerod, V. Neiger, E. Schost, and G. Villard, "On the complexity of multivariate interpolation with multiplicities and of simultaneous polynomial approximations," Presented at ASCM'12, Beijing, China, Oct. 2012.

[13] H. Cohn and N. Heninger, "Ideal forms of Coppersmith's theorem and Guruswami-Sudan list decoding," in *Innovations in Computer Science*. Tsinghua University Press, 2011, pp. 298–308, extended version available at http://arxiv.org/pdf/1008.1284.

[14] ——, "Approximate common divisors via lattices," in *Tenth Algorithmic Number Theory Symposium*. Mathematical Sciences Publishers (MSP), 2012-2013, pp. 271–293.

[15] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *J. Symbolic Comput.*, vol. 9, no. 3, pp. 251–280, 1990.

[16] R. A. DeMillo and R. J. Lipton, "A probabilistic remark on algebraic program testing," *Inform. Process. Lett.*, vol. 7, no. 4, pp. 193–195, 1978.

[17] 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 Trans. Inf. Theory*, vol. 37, no. 5, pp. 1274–1287, 1991.

[18] P. Gaborit and O. Ruatta, "Improved Hermite multivariate polynomial interpolation," in *ISIT'06*. IEEE, 2006, pp. 143–147.

[19] J. Gathen and J. Gerhard, *Modern Computer Algebra (third edition)*. Cambridge University Press, 2013.

[20] P. Giorgi, C.-P. Jeannerod, and G. Villard, "On the complexity of polynomial matrix computations," in *ISSAC'03*. ACM, 2003, pp. 135–142. [Online]. Available: http://doi.acm.org/10.1145/860854.860889

[21] S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote, "Triangular x-basis decompositions and derandomization of linear algebra algorithms over $K[x]$," *J. Symbolic Comput.*, vol. 47, no. 4, pp. 422–453, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jsc.2011.09.006

[22] V. Guruswami and A. Rudra, "Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 135–150, 2008.

[23] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, 1999.

[24] H. Hasse, "Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenem Konstantenkörper bei beliebiger Charakteristik," *J. Reine Angew. Math.*, vol. 175, pp. 50–54, 1936.

[25] E. Kaltofen, "Asymptotically fast solution of Toeplitz-like singular linear systems," in *ISSAC'94*. ACM, 1994, pp. 297–304.

[26] E. Kaltofen and D. Saunders, "On Wiedemann's method of solving sparse linear systems," in *AAECC-9*, ser. LNCS, vol. 539. Springer, 1991, pp. 29–38.

[27] R. Koetter, J. Ma, and A. Vardy, "The re-encoding transformation in algebraic list-decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 633–647, 2011.

[28] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, 2003.

[29] ——, "A complexity reducing transformation in algebraic list decoding of Reed-Solomon codes," in *ITW2003*. IEEE, 2003, pp. 10–13.

[30] R. Kötter, "Fast generalized minimum-distance decoding of algebraic-geometry and Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 721–737, 1996.

[31] F. Le Gall, "Powers of tensors and fast matrix multiplication," in *ISSAC'14*. ACM, 2014, pp. 296–303. [Online]. Available: http://doi.acm.org/10.1145/2608628.2608664

[32] K. Lee and M. E. O'Sullivan, "List decoding of Reed-Solomon codes from a Gröbner basis perspective," *J. Symbolic Comput.*, vol. 43, no. 9, pp. 645–658, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747717108000059

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

[34] H. M. Möller and B. Buchberger, "The construction of multivariate polynomials with preassigned zeros," in *EUROCAM'82*, ser. LNCS, vol. 144. Springer, 1982, pp. 24–31.

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

[36] J. Nielsen, "List decoding of algebraic codes," Ph.D. dissertation, Technical University of Denmark, 2013.

[37] R. R. Nielsen and T. Høholdt, "Decoding Reed-Solomon codes beyond half the minimum distance," in *Coding Theory, Cryptography and Related Areas*. Springer, 2000, pp. 221–236.

[38] V. Olshevsky and M. A. Shokrollahi, "A displacement approach to efficient decoding of algebraic-geometric codes," in *STOC'99*. ACM, 1999, pp. 235–244. [Online]. Available: http://doi.acm.org/10.1145/301250.301311

[39] V. Y. Pan, *Structured Matrices and Polynomials*. Birkhäuser/Springer, Boston/New York, 2001.

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

[41] J.-R. Reinhard, "Algorithme LLL polynomial et applications," Master's thesis, École Polytechnique, Paris, France, 2003, available at https://hal.inria.fr/hal-01101550.

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

[43] R. M. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 246–257, 2000.

[44] S. Sarkar and A. Storjohann, "Normalization of row reduced matrices," in *ISSAC'11*. ACM, 2011, pp. 297–304.

[45] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 701–717, 1980.

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

[47] A. Storjohann, "Notes on computing minimal approximant bases," in *Challenges in Symbolic Computation Software*, ser. Dagstuhl Seminar Proceedings, 2006. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2006/776

[48] A. Stothers, "On the complexity of matrix multiplication," Ph.D. dissertation, University of Edinburgh, 2010.

[49] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *J. Complexity*, vol. 13, no. 1, pp. 180–193, 1997. [Online]. Available: http://dx.doi.org/10.1006/jcom.1997.0439

[50] P. V. Trifonov, "Efficient interpolation in the Guruswami-Sudan algorithm," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4341–4349, 2010.

[51] V. Vassilevska Williams, "Multiplying matrices faster than Coppersmith-Winograd," in *STOC'12*. ACM, 2012, pp. 887–898. [Online]. Available: http://doi.acm.org/10.1145/2213977.2214056

[52] Y. Wu, "New list decoding algorithms for Reed-Solomon and BCH codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3611–3630, 2008.

[53] A. Zeh, "Algebraic Soft- and Hard-Decision Decoding of Generalized Reed–Solomon and Cyclic Codes," Ph.D. dissertation, École Polytechnique, 2013. [Online]. Available: https://pastel.archives-ouvertes.fr/pastel-00866134

[54] A. Zeh, C. Gentner, and D. Augot, "An interpolation procedure for list decoding Reed-Solomon codes based on generalized key equations," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5946–5959, 2011.

[55] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *EUROSAM'79*, ser. LNCS, vol. 72. Springer, 1979, pp. 216–226.

Muhammad F. I. Chowdhury was born in Sylhet, Bangladesh on 31st December 1981. He achieved his BSc in Computer Science and Engineering from Khulna University of Engineering & Technology, Bangladesh. He obtained his MSc in Computer Science from the University of Western Ontario, Canada, on April 2009. In February 2014, he achieved his PhD degree in Computer Science from the University of Western Ontario, Canada. Currently he is working as a senior software engineer at Irdeto Canada Inc. where he is responsible for designing and developing secured algorithms to be executed in untrusted computational environments.

Claude-Pierre Jeannerod received his PhD in Applied Mathematics from Institut National Polytechnique, Grenoble (France), in 2000. After being a postdoctoral fellow in the Symbolic Computation Group at the University of Waterloo (Canada), he is now a researcher at Inria Grenoble - Rhône-Alpes and a member of the LIP Computer Science Laboratory (CNRS, ENSL, Inria, UCBL) of the University of Lyon, France. His research interests include computer algebra, structured linear algebra, and floating-point arithmetic.

Vincent Neiger studied Computer Science at École Normale Supérieure de Lyon in France, where he obtained a Bachelor's degree in 2010 and a Master's degree in 2012, and passed the *Agrégation* national competitive examination in Mathematics in 2013. He is currently working toward a joint PhD degree in Computer Science between École Normale Supérieure de Lyon and the University of Western Ontario, Canada, and is a member of the LIP Computer Science Laboratory (CNRS, ENSL, Inria, UCBL) of the University of Lyon, France, and of the Computer Science Department of the University of Western Ontario, Canada. His current research interests include computer algebra and algebraic coding theory.

Éric Schost received his PhD in Computer Science at École Polytechnique in 2000, under the supervision of Marc Giusti. He is an associate professor in the Department of Computer Science at Western University and holds a Canada Research Chair in Computer Algebra.

Gilles Villard received the PhD degree from the Institut National Polytechnique of Grenoble, and became a research scientist with the French National Center for Scientific Research (CNRS) in 1990. He arrived at the École Normale Supérieure de Lyon in 2000 and has headed the project-team Arénaire on Computer Arithmetic between 2004 and 2009. He has been vice-chair then chair of the LIP Computer Science Laboratory (CNRS, ENSL, Inria, UCBL) of the University of Lyon from 2006 to 2014. His main research interests in symbolic computation are complexity and efficient algorithms for matrix and Euclidean lattice problems, and generic programming techniques for high performance software libraries.