MATHEMATICS OF COMPUTATION
Volume 77, Number 261, January 2008, Pages 447–473
S 0025-5718(07)02012-1
Article electronically published on September 13, 2007

CORRELATED ALGEBRAIC-GEOMETRIC CODES: IMPROVED LIST DECODING OVER BOUNDED ALPHABETS

VENKATESAN GURUSWAMI AND ANINDYA C. PATTHAK

Abstract. We define a new family of error-correcting codes based on algebraic curves over finite fields, and develop efficient list decoding algorithms for them. Our codes extend the class of algebraic-geometric (AG) codes via a (nonobvious) generalization of the approach in the recent breakthrough work of Parvaresh and Vardy (2005).

Our work shows that the PV framework applies to fairly general settings by elucidating the key algebraic concepts underlying it. Also, more importantly, AG codes of arbitrary block length exist over *fixed* alphabets Σ, thus enabling us to establish new trade-offs between the list decoding radius and rate over a bounded alphabet size.

The work of Parvaresh and Vardy (2005) was extended in Guruswami and Rudra (2006) to give explicit codes that achieve the list decoding capacity (optimal trade-off between rate and fraction of errors corrected) over large alphabets. A similar extension of this work along the lines of Guruswami and Rudra could have substantial impact. Indeed, it could give better trade-offs than currently known over a fixed alphabet (say, $\mathrm{GF}(2^{12})$), which in turn, upon concatenation with a fixed, well-understood binary code, could take us closer to the list decoding capacity for binary codes. This may also be a promising way to address the significant complexity drawback of the result of Guruswami and Rudra, and to enable approaching capacity with bounded list size independent of the block length (the list size and decoding complexity in their work are both $n^{\Omega(1/\varepsilon)}$ where ε is the distance to capacity).

Similar to algorithms for AG codes from Guruswami and Sudan (1999) and (2001), our encoding/decoding algorithms run in polynomial time assuming a natural polynomial-size representation of the code. For codes based on a specific "optimal" algebraic curve, we also present an expected polynomial time algorithm to construct the requisite representation. This in turn fills an important void in the literature by presenting an efficient construction of the representation often assumed in the list decoding algorithms for AG codes.

1. Introduction

In this work, we define a new family of algebraic codes and develop list decoding algorithms for them. These codes are obtained by generalizing the approach of

Received by the editor July 26, 2006 and, in revised form, November 14, 2006.
2000 *Mathematics Subject Classification.* Primary 94B27, 12Y05, 14Q05, 14H05.
An extended abstract describing some of these results was presented at the 47th Annual Symposium on Foundations of Computer Science (FOCS), 2006. This is an expanded version of the paper, containing the proofs and algorithms in full.
The first author was supported by NSF Career Award CCF-0343672, an Alfred P. Sloan Research Fellowship, and a David and Lucile Packard Foundation Fellowship.
The second author was supported in part by NSF Grant CCR-0310960.

Parvaresh and Vardy [11], which applied to Reed-Solomon (RS) codes to algebraic-geometric (AG) codes. Below we describe the context and motivation for our work followed by a description of some of our results.

1.1. **Context and motivation.** The basic trade-off underlying the theory of error-correcting codes is the one between the amount of noise the codes can handle (the error-correction radius) and the amount of redundancy the coding scheme introduces. The latter is measured by the *rate* of the code, which is defined as the number of information symbols to the length of the encoding (called *block length*). In this work, we focus on *worst-case* errors, where we assume a bound on the fraction of errors the channel may effect, but nothing about how the error locations or values are distributed.

Suppose we encode messages with $\mathcal{R}n$ symbols of information over an alphabet Σ into codewords of n symbols over Σ (here $\mathcal{R}$ is the rate; we think of $\mathcal{R}$ as an absolute constant and let the block length $n \to \infty$). Clearly, to recover the $\mathcal{R}n$ message symbols, we need at least $\mathcal{R}n$ correct symbols at the receiving end. Thus, the absolute information-theoretic limit on a fraction of correctable errors is $1-\mathcal{R}$. Surprisingly, a notion called list decoding offers the potential to approach this limit (called "capacity"). Under list decoding up to a fraction p of errors, the decoder is required to output a list of all codewords which differ from the received word in at most a fraction p of symbols. The *list size* L needed for list decoding is the maximum number of codewords that are output in the worst-case. In the limit of $L \to \infty$, there exist list decodable codes of rate $\mathcal{R}$ that can be decoded up to the information-theoretically optimal $1-\mathcal{R}$ fraction of errors. We remark that this is *twice* the fraction of errors that can be corrected with unique decoding (the case $L = 1$).

The above, however, is a nonconstructive result. The codes achieving list decoding capacity were random codes with decoding algorithms no better than exponential-time brute-force search (this is akin to the codes in Shannon's original work for stochastic channels). Recently, building on a line of work in algebraic coding theory [14, 6, 11], explicit codes (called folded Reed-Solomon codes) that achieve list decoding capacity with polynomial encoding/decoding complexity were given in [4].

The work of [4] thus meets the challenge of achieving capacity for worst-case errors. However, it has some drawbacks relating to complexity. To correct a fraction $(1-\mathcal{R}-\varepsilon)$ of errors, the proven bound on the worst-case list size of the algorithm in [4] is $n^{\Omega(1/\varepsilon)}$. In contrast, the existential result gets within ε of capacity with list size $O(1/\varepsilon)$. It is an important goal to improve the list size to a constant independent of n. The dependence of the list size on n in [4] arises because Reed-Solomon codes need an alphabet of size at least n. This motivates one to generalize this approach to AG codes which can have arbitrary block lengths over fixed alphabets, and also have very nice algebraic properties. Recent advances have greatly improved the efficiency and explicitness of constructions of AG codes [12], making this a promising route to approach capacity with better list size and decoding complexity.

The codes in [4] are defined over a large alphabet (of size $2^{O(1/\varepsilon^4)}$ to get within ε of capacity). For codes over alphabet size q for a fixed bounded constant q (say $q = 2^{12}$), the best general trade-off for error correction radius vs. rate remains the $(1-1/q)(1-\sqrt{\mathcal{R}})$ bound obtained in [6, 9] for AG codes. Improving this state of affairs provides another motivation for extending the Parvaresh-Vardy approach to AG codes.

1.2. **Our contribution.** Motivated by the above concerns, in this work we present a generalization of the Parvaresh-Vardy approach to all AG codes. To describe our contribution, we begin with the variant of RS codes that was put forth in [11]. In an RS code, the message is a polynomial, which is encoded by its evaluations at elements of a field. In the PV-scheme, the message is a polynomial f, and then a related polynomial h is computed (as a carefully chosen function of f – the details of how this is done are crucial to the success of this approach), and then the encoding comprises of the evaluations of both f and h on the field elements. This gives a nonlinear code of half the rate compared to the original RS code. To get better trade-offs between rate and list decoding radius for very low rates, one can use not a pair but an M-tuple of correlated polynomials for the encoding.

In this work, we generalize this approach to all AG codes, and define the class of correlated AG codes, based on evaluations of correlated functions from a suitable linear space at points on an algebraic curve. This highlights the generality and promise of the Parvaresh-Vardy approach, and elucidates its salient features in a general setting unencumbered by specifics of a particular code. Some of the challenges in such a generalization are discussed in Section 2.

We now describe some of the new trade-offs for list decoding our work implies. For q an even power of a prime, and any integer $m \geqslant 1$, we present codes with rate $\mathcal{R}$ and list decoding radius approximately $1 - (m\mathcal{R} + 3/\sqrt{q})^{m/(m+1)}$ over an alphabet of size q^m. (Here m is the number of correlated functions used for the encoding, and so $m = 1$ corresponds to list decoding the usual AG codes.) For low rates $\mathcal{R}$ and large values of m, this gives an improvement over the trade-off $1-(\mathcal{R}+1/\sqrt{q})^{1/2}$ for the usual AG codes (the $m = 1$ case). In particular, for small $\varepsilon \to 0$, we can correct up to a fraction $(1-\varepsilon)$ of errors with rate $\Omega(\varepsilon/\log(1/\varepsilon))$ and alphabet size $2^{O(\log^2(1/\varepsilon))}$. Contrast this with the existential result showing that one can list decode to a radius of $(1-\varepsilon)$ with rate $\Omega(\varepsilon)$ and alphabet size $O(1/\varepsilon^2)$.

Our decoding algorithms run in polynomial time assuming a polynomial sized preprocessed representation of the code. With a slight weakening of the error-correction performance, we present a different algorithm for which the preprocessing can also be done in polynomial time. These issues concerning polynomial runtime are further clarified in Section 1.3.

Previously the only polynomial time constructions for decoding up to radius $(1-\varepsilon)$ with alphabet size $\mathrm{poly}(1/\varepsilon)$ achieved rate $\Omega(\varepsilon^2)$ (this follows from the list decoding of AG codes in [6]). Our results give the **first codes** with rate better than $\Omega(\varepsilon^2)$, say $\Omega(\varepsilon^{1.1})$, over an alphabet of size polynomial in $1/\varepsilon$. Thus, our result does well simultaneously on both the alphabet size vs. list decoding radius and the rate vs. list decoding radius trade-offs.

Our codes also have a nice *list recovering* property which can be used in concatenation schemes with suitable *constant-sized* inner codes to get the first uniformly constructive binary codes of rate close to ε^3 list-decodable up to radius $(1/2-\varepsilon)$ with list size depending only on ε and independent of n. (The construction in [4] with a similar rate needed construction time of the form $n^{f(\varepsilon)}$ instead of the $f(\varepsilon)n^{O(1)}$ we achieve, and their list size also depends on n.)

Guruswami and Rudra [4] extended the work of Parvaresh and Vardy by arranging for the correlated polynomials to be just the original polynomial with a "shift". This was the algebraic crux of their work. At this point, we do not know how to extend this work for correlated AG codes along similar lines. Such an extension is

an important direction of future work with potentially significant impact. Indeed, it would be a promising way to address the significant complexity drawback of the result in [4], and to enable approaching capacity with bounded list size independent of the block length. Moreover, it could yield codes with significantly improved trade-offs over a fixed alphabet, which upon concatenation with a well-understood constant-sized binary code (e.g., see [8]), could take us closer to the challenging goal of achieving list decoding capacity for binary codes.

1.3. **Complexity of encoding/decoding.** Since AG codes are a whole family of codes as opposed to a specific code, when we say we give polynomial time encoding and decoding algorithms for them, we mean that every AG-code has a representation of polynomial size given which there are encoding/decoding procedures that run in polynomial time. This is similar to the situation for the original list decoding algorithm for AG codes [6, 7], and is the best one can hope for when we want to decode *every* AG code of a certain type.

However, it makes sense to try to construct this requisite representation efficiently for certain specific AG-codes, ideally the ones which offer the best trade-offs for list decoding. We explicitly address this question in Section 7. For the specific "optimal" AG codes based on a tower of function fields due to Garcia and Stichtenoth [1, 2], we give an expected polynomial time (i.e., *Las Vegas*) construction of the description of the code needed for our algorithms. Though not explicit in the sense of deterministic polynomial time constructibility, the representation is guaranteed to be correct and constructing it (a one time job) takes polynomial time with overwhelming probability. This level of explicitness should thus suffice for using the code. We remark that even for the algorithm of Guruswami and Sudan [6, 7] (that achieved a decoding radius of at most $1 - \sqrt{\mathcal{R}}$), it was not known how to compute the required representation efficiently. Our construction thus fills an important void in the literature on efficient decoding of AG codes, and we view this also as an important algorithmic contribution of this work.

1.4. **Organization.** We describe some of the hurdles that need to be overcome in generalizing the PV framework to AG codes in Section 2. Some basic definitions and terminology concerning AG codes and function fields are discussed in Section 3. We describe our actual code construction in Section 4. We describe the first of our decoding algorithms in Section 5, and a second decoding algorithm with a better error-correction performance in Section 6. In Section 7, we prove that for certain "optimal" AG codes certain preprocessed information needed by our algorithms can be computed in expected polynomial time. While this preprocessed information suffices for the first algorithm to run in polynomial time, the second algorithm needs further preprocessed information about the code to run in polynomial time, but we do not know how to compute the additional preprocessed information it needs efficiently. We describe extensions to the problem of list recovering and constructions of binary codes for list decoding up to a fraction $(1/2 - \varepsilon)$ of errors in Section 8.

2. Generalizing to AG-codes: Ideas and complications

As mentioned above, in this work we propose a generalization of the Parvaresh-Vardy coding scheme to AG codes. While fairly natural in hindsight (which a

"correct" generalization ought to be!), the generalization to AG codes is not immediate, since, as we describe below, the special structure of RS codes and the rational function field $\mathbb{F}_q(X)$ are used in a more than superficial way in [11].

The ability to view a low-degree polynomial (i.e., the function being evaluated) also as a field element from some field $\mathbb{F}$, and operating on it in the field $\mathbb{F}$ to get another related polynomial is crucial to the PV construction. Indeed, the decoding is performed by solving a system of polynomial equations over the field $\mathbb{F}$ whose solutions contain all possible codewords that must be output. For Reed-Solomon codes, there is a natural way to view polynomials as field elements, since polynomials of degree $< k$ are in one-to-one correspondence with elements of the extension field $\mathbb{F}_q[X]/(E(X)) \approx \mathbb{F}_{q^k}$ (where $E(X)$ is an irreducible polynomial of degree k over $\mathbb{F}_q$). In order to generalize this framework to AG codes, we need an injective homomorphism from the elements of the function field K that are evaluated to give the AG-encoding (i.e., the analog of low-degree polynomials for the RS case) to a suitable field $\mathbb{F}$. We achieve this by associating with an element f of the function field, the element of the field $\mathbb{F}_{q^\alpha}$ which is the evaluation $f(R)$ of f at a fixed place R of (large enough) degree α. This evaluation is then used to obtain, from the message function f, a correlated function h such that $h(R)$ is a carefully chosen function of $f(R)$. Unlike the RS case, however, for function fields of larger genus this evaluation map restricted to the message functions is only injective and not bijective.[1] Fortunately, we are able to show (Lemma 4.2) that a correlated function h with the desired evaluation $h(R)$ always exists in a slightly larger space compared to the message space to which f belongs.

The decoding algorithm follows the interpolation followed by root-finding idea that is common to [14, 6, 11]. However, another technical complication arises in the phase when the interpolated polynomial, say Q, is mapped into a polynomial N with coefficients from $\mathbb{F}_{q^\alpha}$ by evaluating each of its coefficients at the place R. Following [11], we seek to find roots in $\mathbb{F}_{q^\alpha}$ of N, and using the above-mentioned injection from messages into $\mathbb{F}_{q^\alpha}$, map these roots back to obtain the list of messages. It is crucial that in this step N is a nonzero polynomial when Q is. For the Reed-Solomon case, this is easy to achieve, since the coefficients of Q, which are polynomials over $\mathbb{F}_q$ in one variable, come from a *principal ideal domain* (PID), i.e., a ring all of whose ideals are generated by a single element. Therefore, the only way N can be zero when Q is nonzero is if all coefficients of Q are divisible by the generator of the ideal R (i.e., by a univariate polynomial $E(X)$ of degree α). In this case we can divide Q by the appropriate power of $E(X)$ to get a lower-degree nonzero polynomial $\tilde{Q}$ which is not divisible by $E(X)$, and then work with it instead.

However, for general function fields, the ring $\emptyset$ to which the coefficients of Q typically belong is not a PID. Therefore, even if all coefficients of Q vanish at R, they may not share a common factor in $\emptyset$ and the above approach for RS codes cannot be applied. We circumvent this issue in two ways giving two different algorithms, each with its own advantages, as described below.

In the first approach, we restrict the coefficients of Q to come from a much smaller space of functions than is usually done in the interpolation based algorithms of [14, 6, 11]. Specifically, we restrict the pole order of each of the functions to be

[1] A bijective map can be shown to exist provided a general divisor, instead of a divisor supported on one point, is chosen to define the code. However, we do not know how to compute this divisor efficiently.

less than α. This ensures that no nonzero coefficient of Q evaluates to 0 at the place R, which has degree α. Therefore, $Q \neq 0$ implies $N \neq 0$, as desired. This restriction on the coefficients of Q does not come for free, however, and we need to give up a bit on the potential performance in terms of number of errors corrected. In particular, this approach begins to give improvements over the decoding of regular AG-codes only when we use 3 or more correlated functions (as opposed to the case of RS codes in [11], where a pair of functions already gives a substantial improvement). Another fall out of our stringent restriction on the coefficients of Q is that the idea of using *large* "multiplicities" in the interpolation phase actually *degrades* the error correction performance of the algorithm (it does gives minor improvements for small multiplicities, the best one being for multiplicity 3 for the case of three correlated functions). This is in contrast to [6, 11, 4] where large multiplicities are crucial for the claimed performance. On the flip side, this greatly helps us in Section 7 since the construction of the requisite representation of the AG code is simpler when one does not have to deal with multiplicities. The advantage of this approach is thus its simplicity — the decoding algorithm needs the *same* representation of the code as the encoding, and this representation can be computed in (expected) polynomial time.

In the second approach, we do not impose additional restrictions on the coefficients of Q beyond the usual interpolation based algorithms. Instead, if all coefficients of Q vanish at R, we multiply each of the coefficients of Q by a function ν^c where ν is a function with a pole of order 1 at R and no poles elsewhere (such a function must exist if the degree of R is large), and $c \geqslant 1$ is the minimum of the zero orders at R of the coefficients of Q. We then reduce the resulting polynomial $\tilde{Q} = \nu^c Q$ modulo R to get a nonzero polynomial N with coefficients in $\mathbb{F}_{q^\alpha}$ and then proceed as before. Several challenges arise in implementing this idea. First, we need a way to represent ν and a way to compute c. Also, the coefficients of $\tilde{Q}$ are no longer in the ring $\emptyset$, making it difficult to represent and evaluate them efficiently. Nevertheless, we prove that the coefficients of $\tilde{Q}$ belong to a linear space of functions with bounded number of poles at R. We use this to compute c as well as a representation of the coefficients of $\tilde{Q}$ that lets us evaluate them at R (assuming some extra preprocessed information). The advantage of this approach is that we can use large multiplicities in the interpolation phase and as a result there is no degradation in error-correction radius compared to the results of Parvaresh-Vardy (for example, using two correlated functions already suffices to go beyond regular AG codes). The drawback is that the decoding algorithm needs more complicated, albeit still polynomial amount of preprocessed information, and we do not know how to perform the pre-processing in polynomial time (but given the preprocessed information, the algorithm runs in polynomial time).

3. Background on algebraic-geometric codes

Most of the notation and terminology we use is standard in the study of algebraic-geometric codes, and can be found in Stichtenoth's book [13]. We briefly recapitulate some key facts concerning algebraic function fields and algebraic-geometric codes that we need for our description. Let K be a function field over $\mathbb{F}_q$, denoted $K/\mathbb{F}_q$, i.e., a finite algebraic extension of the field $\mathbb{F}_q(X)$ of rational functions over $\mathbb{F}_q$. A subring X of K is said to be a valuation ring if for every $z \in K$, either $z \in X$ or $z^{-1} \in X$. Each valuation ring is a *local ring*, i.e., it has a unique maximal ideal.

The set of places of K, denoted $\mathbb{P}_K$, is the set of maximal ideals of all the valuation rings of K. Geometrically, this corresponds to the set of all (nonsingular) points on the algebraic curve corresponding to K. The valuation ring corresponding to a place P is called the ring of regular functions at P and is denoted $\emptyset_P$. Associated with a place P is a valuation $v_P : K \to \mathbb{Z} \cup \{\infty\}$, that measures the order of zeroes or poles of a function at P (with the convention $v_P(0) = \infty$). In terms of v_P, we have $\emptyset_P = \{x \in K \mid v_P(x) \geqslant 0\}$ and $P = \{x \in K \mid v_P(x) > 0\}$. The quotient $\emptyset_P/P$ is a field since P is a maximal ideal – it is called the residue field at P. The residue field $\emptyset_P/P$ is a finite extension field of $\mathbb{F}_q$; the degree of this extension is called the *degree* of P, and is denoted $\deg(P)$. For every place P, we have an evaluation map $\mathsf{ev}_P : \emptyset_P \to \emptyset_P/P$ defined by $\mathsf{ev}_P(z) = z(P) = z + P$; this map is $\mathbb{F}_q$-linear. We will think of ev_P as a map into $\mathbb{F}_{q^{\deg(P)}}$ using an isomorphism of the residue field to $\mathbb{F}_{q^{\deg(P)}}$. Thus, elements of K can be viewed as functions on $\mathbb{P}_K$ (hence the name *function field* for K); the evaluation of $z \in K$ and $P \in \mathbb{P}_K$, denoted $z(P)$, is either ∞ (if $z \notin \emptyset_P$) or belongs to $\mathbb{F}_{q^{\deg(P)}}$.

The set of divisors D_K of a function field $K/\mathbb{F}_q$ is the (additively written) free abelian group generated by the places $\mathbb{P}_K$. For a divisor $D = \sum_{P \in \mathbb{P}_K} n_P P$ where all but finitely many n_P are 0, its degree, denoted $\deg(D)$, is defined as $\deg(D) = \sum_{P \in \mathbb{P}_K} n_P \deg(P)$ (note that this is a finite sum). For a divisor $D = \sum_P n_P P$, we define the set of functions $Ł(D) \stackrel{\text{def}}{=} \{x \in K \mid v_P(x) \geqslant -n_P \ \forall P \in \mathbb{P}_K\}$; this forms a vector space over $\mathbb{F}_q$.

Theorem 3.1 (Follows from Riemann-Roch)**.** *If $D \in D_K$ is a divisor of $K/\mathbb{F}_q$ of degree at least $2g-1$, then* $\dim(L(D)) = \deg(D) - g + 1$.

An algebraic-geometric code over $\mathbb{F}_q$ is obtained by evaluating a carefully chosen subset of elements of K at *places of degree one*. For a place P_∞ of degree one and an integer α, the set $Ł((\alpha-1)P_\infty)$ consists of all those $z \in K$ for which z has no poles at places other than P_∞, and may have a pole at P_∞ of order less than α. Typically, an AG-code is defined to be the evaluations of functions in $Ł((\alpha-1)P_\infty)$ at n distinct places $P_1, P_2, \ldots, P_n$ (different from P_∞) of degree one. That is,

$$C_{\alpha,P_\infty} = \{\langle f(P_1), f(P_2), \ldots, f(P_n)\rangle \mid f \in Ł((\alpha-1)P_\infty)\} \ .$$

This is a linear code since $Ł((\alpha-1)P_\infty)$ is a vector space over $\mathbb{F}_q$. The dimension of C_{α,P_∞} is at least $\alpha - g$ by the Riemann-Roch theorem. Its minimum distance is at least $n - \alpha + 1$ since a nonzero function in $Ł((\alpha-1)P_\infty)$ can have at most $(\alpha-1)$ zeroes.

4. Construction of correlated AG codes

We now describe a correlated AG code construction where we use a **triple** of functions in the evaluation. As mentioned above, our scheme does not get an improvement in decoding performance (compared to regular AG-codes) when just two correlated functions are used for the evaluation. The extension of the code, decoding algorithm, and analysis for the case when more than three correlated functions are evaluated as part of encoding, follows in a natural way, and are discussed briefly in Section 5.6.

We now describe our construction of the code. Most of the notation and terminology we use is standard in the study of algebraic-geometric codes, and can be

found in Stichtenoth's book [13]. Let K be a function field over $\mathbb{F}_q$ corresponding to a smooth, irreducible curve. Let g be the genus of K. Suppose K has at least $n+1$ places of degree one, say $P_1, \ldots, P_n$ and P_∞. Let $k \geqslant g$ be arbitrary (this assumption is mainly for convenience). We will describe a code C of block length n over alphabet $\mathbb{F}_{q^3}$ with q^k codewords. The rate of the code will thus be $r(C) = k/(3n)$. The code will **not** be linear. Let $\{1, \beta_1, \beta_2\}$ be a basis of $\mathbb{F}_{q^3}$ over $\mathbb{F}_q$.

The messages of C will be identified with the vector space $\mathbb{F}_q^k$. We specify the code by specifying its encoding function, Enc, which will be an injective map $\mathsf{Enc} : \mathbb{F}_q^k \to \left(\mathbb{F}_{q^3}\right)^n$.

Let $\alpha = k + g$. We denote by $Ł((\alpha-1)P_\infty)$ the set of functions in K that have no poles outside the place P_∞ and may have a pole at P_∞ of order less than α. Since $\alpha - 1 \geqslant 2g - 1$, by the Riemann-Roch theorem, $Ł((\alpha-1)P_\infty)$ is a k-dimensional vector space over $\mathbb{F}_q$, and it is with this space that we identify our messages. Let $\phi_1, \phi_2, \ldots, \phi_k$ be a basis of $Ł((\alpha-1)P_\infty)$. Specifically, a message $(a_1, a_2, \ldots, a_k) \in \mathbb{F}_q^k$ will be viewed as the element $a_1\phi_1 + \cdots + a_k\phi_k \in Ł((\alpha-1)P_\infty)$. Therefore, we will describe our encoding function as a map

$$\mathsf{Enc} : Ł((\alpha-1)P_\infty) \to (\mathbb{F}_{q^3})^n \ . \tag{1}$$

It is well known that for every place P, we have an evaluation map $\mathsf{ev}_P : Ø_P \to Ø_P/P$ defined by $\mathsf{ev}_P(z) = z(P) = z + P$; this map is $\mathbb{F}_q$-linear. Let $R \in \mathbb{P}_K$ be a place of degree α.[2] We begin with the following simple lemma, which lets us view our messages as a subset of $\mathbb{F}_{q^\alpha}$, using their evaluations at R. Note that $Ł((\alpha-1)P_\infty) \subseteq Ø_R$ since functions in $Ł((\alpha-1)P_\infty)$ have no poles outside P_∞ and thus certainly do not have a pole at R.

Lemma 4.1. *The restriction of the map* ev_R *to* $Ł((\alpha-1)P_\infty)$ *is injective. Its range is a* k*-dimensional subspace of* $\mathbb{F}_{q^\alpha}$*.*

Proof. Indeed, if $f_1, f_2 \in Ł((\alpha-1)P_\infty)$ satisfy $f_1(R) = f_2(R)$, then $f_1 - f_2$ has a zero at R. Hence the zero divisor of $f_1 - f_2$ has degree at least $\deg(R) = \alpha$. However, the pole divisor of $f_1 - f_2$ has degree at most $\alpha - 1$ since $f_1 - f_2 \in Ł((\alpha-1)P_\infty)$. Therefore we must have $f_1 - f_2 = 0$. Since ev_R is $\mathbb{F}_q$-linear, and $Ł((\alpha-1)P_\infty)$ is a k-dimensional vector space over $\mathbb{F}_q$, the image $\mathsf{ev}_R(Ł((\alpha-1)P_\infty))$ is a k-dimensional subspace. □

Our plan is to use the above as follows. We can view the message

$$f \in Ł((\alpha-1)P_\infty)$$

as the field element $f(R)$. We can attempt to define a correlated message h whose evaluation $h(R)$ is an appropriate function Γ (over $\mathbb{F}_{q^\alpha}$) applied to $f(R)$.[3] However, for the decoding procedure, it seems important that this function Γ be nonlinear (over $\mathbb{F}_q$). The image of ev_R restricted to $Ł((\alpha-1)P_\infty)$ is a subspace of $\mathbb{F}_{q^\alpha}$. When Γ is not linear, in general there may not exist $h \in Ł((\alpha-1)P_\infty)$ satisfying

[2] We note that a place of degree d exists for all d such that $(q^d - 1) > 2q^{d/2}g$, and $d = \alpha \geqslant 2g$ satisfies this condition.

[3] More generally, following the PV-scheme, we can let $(f(R), h(R))$ belong to some curve, but this will improve parameters slightly at best.

$h(R) = \Gamma(f(R))$.[4] The following crucial lemma shows that such an h exists provided we allow slightly bigger pole order at P_∞.

Lemma 4.2. *The image of* $L((\alpha + 2g - 1)P_\infty)$ *under* ev_R *equals* $\mathbb{F}_{q^\alpha}$.

Proof. Let D be the divisor $(\alpha + 2g - 1)P_\infty$. We wish to show that the restriction of ev_R to $L(D)$, denote it by $H : L(D) \to \mathbb{F}_{q^\alpha}$, is surjective. Note that H is an $\mathbb{F}_q$-linear map, so the image of H, $\mathsf{Im}(H)$, is a subspace of $\mathbb{F}_{q^\alpha}$. We will show that $\mathsf{Im}(H)$ has dimension α, and this will show that H is surjective.

The image $\mathsf{Im}(H)$ is isomorphic to the quotient $L(D)/\ker(H)$ where $\ker(H)$ is the kernel of H. Recalling that $H(z) = 0$ iff $\mathsf{ev}_R(z) = 0$, we have $\ker(H) = \{z \in L(D) \mid z \text{ has a zero at } R\}$. Thus we have $\ker(H) = L(D - R)$. Therefore, $\dim(\mathsf{Im}(H)) = \dim(L(D)/L(D - R)) = \dim(L(D)) - \dim(L(D - R))$.

Now, by the Riemann-Roch theorem, $\dim(L(D)) = (\alpha + 2g - 1) - g + 1 = \alpha + g$, and $\dim(L(D - R)) = \deg(D - R) - g + 1 = ((\alpha + 2g - 1) - \alpha) - g + 1 = g$. It follows that $\dim(\mathsf{Im}(H)) = \alpha$, as desired. □

Before we finally describe the encoding function, we need one other notation. For each $\gamma \in \mathbb{F}_{q^\alpha}$, we fix an arbitrary preimage in $L((\alpha+2g-1)P_\infty)$, denote it $I[\gamma]$, that satisfies $\mathsf{ev}_R(I[\gamma]) = \gamma$. (Such a preimage exists by Lemma 4.2.) The code will be parameterized by integers $s_1, s_2 \geqslant 1$ (which will be specified later when we analyze the decoding algorithm). For $f \in L((\alpha - 1)P_\infty)$, we define the ith coordinate of $\mathsf{Enc}(f)$, for $i = 1, 2, \ldots, n$, by

$$\mathsf{Enc}(f)_i = f(P_i) \ + \ \beta_1 \cdot I[f(R)^{s_1}](P_i) \ + \ \beta_2 \cdot I[f(R)^{s_2}](P_i) \tag{2}$$

(recall that $\{1, \beta_1, \beta_2\}$ is a basis of $\mathbb{F}_{q^3}$ over $\mathbb{F}_q$). In other words, the encoding consists of the evaluation $f(P_i)$ and also the evaluations $h_1(P_i)$ and $h_2(P_i)$ where h_i is a specific function that satisfies $h_i(R) = f(R)^{s_i}$ for $i = 1, 2$ (the raising to the s_ith power happens in the field $\mathbb{F}_{q^\alpha}$).

Parameters. Note that the rate of C is $k/(3n)$ and its distance d is at least $n - \alpha + 1 = n - k - g + 1$. Its alphabet size is q^3.

Encoding complexity. The above encoding can be performed in polynomial time, provided (i) we can efficiently compute functions in $L((\alpha+2g-1)P_\infty)$ at the places $P_1, P_2, \ldots, P_n$ and R, and (ii) we can compute the preimage $I[\gamma] \in L((\alpha+2g-1)P_\infty)$ of arbitrary $\gamma \in \mathbb{F}_{q^\alpha}$ efficiently. Since the space $L((\alpha + 2g - 1)P_\infty)$ is an $(\alpha + g)$-dimensional $\mathbb{F}_q$-vector space, both of these tasks can be solved in polynomial time using elementary linear algebra, assuming we have a basis for $L((\alpha + 2g - 1)P_\infty)$ together with the evaluations of the basis functions at P_i as well as at R. This is the representation which we assume for our code, and in Section 7 we will describe how to construct this representation for a specific family of AG codes.

5. Interpolation based decoding: The first algorithm

We now turn to list decoding the above code construction. We recollect the notation of relevant parameters in the construction:

- block length n;
- places $P_1, \ldots, P_n, P_\infty$ of degree 1;

[4] For the Reed-Solomon case, $g = 0$, and hence $\alpha = k$ and so the image $\mathsf{ev}_R(L((\alpha - 1)P_\infty)) = \mathbb{F}_{q^\alpha}$, and so such an $h \in L((\alpha - 1)P_\infty)$ satisfying $h(R) = \Gamma(f(R))$ will always exist.

- message length k (over $\mathbb{F}_q$); $\alpha = k + g$; messages correspond to functions in $\mathrm{L}((\alpha-1)P_\infty)$;
- a place R of degree α;
- the powering exponents s_1, s_2 (these will be specified later).

The list decoding problem for radius $n - t$ amounts to solving the following function reconstruction problem:

Input: Triples $(y_i, z_{1i}, z_{2i}) \in \mathbb{F}_q^3$, for $i = 1, 2, \ldots, n$.

Output: All functions $f \in \mathrm{L}((\alpha-1)P_\infty)$ for which the triple of functions $(f,\ h_1 = I[f(R)^{s_1}],\ h_2 = I[f(R)^{s_2}])$ satisfies $f(P_i) = y_i$, $h_1(P_i) = z_{1i}$ and $h_2(P_i) = z_{2i}$ for at least t values of $i \in \{1, 2, \ldots, n\}$.

5.1. **High level idea behind the algorithm.** Let A denote the ring $\bigcup_{\ell \geqslant 0} \mathrm{L}(\ell P_\infty)$ of all functions in K that have no poles other than possibly at P_∞. The basic idea, following the interpolation based decoding procedure of [14, 6, 11], is to find a nonzero polynomial Q in the polynomial ring $A[Y, Z_1, Z_2]$ such that all triples (f, h_1, h_2) that meet the above output condition are roots of Q. The properties we would like from the interpolation polynomial $Q \in A[Y, Z_1, Z_2]$ are as follows (here ℓ is a suitable integer parameter):

(1) Q is nonzero.
(2) For all $f, h_1, h_2 \in \mathrm{L}((\alpha + 2g - 1)P_\infty)$, $Q(f, h_1, h_2) \in \mathrm{L}((\ell - 1)P_\infty)$.[5]
(3) For every $i = 1, 2, \ldots, n$, for all (f, h_1, h_2) which satisfy $f(P_i) = y_i$, $h_1(P_i) = z_{1i}$ and $h_2(P_i) = z_{2i}$, $Q(f, h_1, h_2)$ has a zero at P_i.

Such a Q can be found in the same way as in [6] (except even simpler, since we only insist on simple zeroes and not zeroes of higher multiplicities), by finding a nonzero solution to an appropriate homogeneous linear system over $\mathbb{F}_q$. The following simple lemma shows the utility of such a polynomial Q.

Lemma 5.1. *Let Q satisfy the above conditions. Let $f, h_1, h_2 \in \mathrm{L}((\alpha+2g-1)P_\infty)$ satisfy $f(P_i) = y_i$, $h_1(P_i) = z_{1i}$ and $h_2(P_i) = z_{2i}$ for t values of i. If $t \geqslant \ell$, then $Q(f, h_1, h_2) = 0$.*

Proof. The function $Q(f, h_1, h_2)$ has at most $(\ell - 1)$ poles, and it has a zero at P_i for each i for which $f(P_i) = y_i$, $h_1(P_i) = z_{1i}$ and $h_2(P_i) = z_{2i}$, and thus at least t zeroes. If $t \geqslant \ell$, this implies that $Q(f, h_1, h_2) = 0$. □

However, once such a Q is found, it will have exponentially many roots in general, so finding all of them and looking for valid triples (f, h_1, h_2) among them is not an option. Instead, we reduce the polynomial Q modulo the place R, by evaluating each of its coefficients at R, to obtain a polynomial $N \in \mathbb{F}_{q^\alpha}[Y, Z_1, Z_2]$. At this step, as mentioned earlier, we have to be careful that N remains a nonzero polynomial.

If (f, h_1, h_2) is a root of Q, clearly the evaluation $(f(R), h_1(R), h_2(R))$ is a root of N. This together with the fact that $h_i(R) = f(R)^{s_i}$ for $i = 1, 2$ implies that $f(R)$ is a root of the univariate polynomial $N(Y, Y^{s_1}, Y^{s_2})$, call it $T(Y)$. By Lemma 4.1, the message $f \in \mathrm{L}((\alpha-1)P_\infty)$ is uniquely recoverable from its evaluation $f(R)$, and so all the solution messages f (and hence the triples (f, h_1, h_2)) can be found by checking amongst the roots of the polynomial T. One additional point to be careful

[5] It will actually suffice for us to require that $Q(f, h_1, h_2) \in \mathrm{L}((\ell - 1)P_\infty)$ whenever $f \in \mathrm{L}((\alpha - 1)P_\infty)$ and $h_1, h_2 \in \mathrm{L}((\alpha + 2g - 1)P_\infty)$. But for sake of uniformity and simplicity, we ensure this also for f from the larger space $\mathrm{L}((\alpha + 2g - 1)P_\infty)$.

about is that T does not become the zero polynomial (even though $N(Y, Z_1, Z_2)$ is nonzero). This is ensured by a suitable, large enough choice of s_1, s_2.

5.2. **Formal description of the decoding algorithm.** We now specify the decoding algorithm outlined in Section 5.1 formally. Recall that the input to the decoding algorithm is a string consisting of triples $(y_i, z_{1i}, z_{2i}) \in \mathbb{F}_q^3$, and the algorithm should find all codewords with agreement t or more with the input string (the parameter t will come out from the analysis). In what follows, $\phi_1, \phi_2, \ldots, \phi_k$ denotes a basis of $\text{Ł}((\alpha-1)P_\infty)$ (recall that this is a k-dimensional vector space over $\mathbb{F}_q$).

Step 0: Compute integer parameters p, ℓ where

$$\ell \stackrel{\text{def}}{=} p(\alpha + 2g - 1) + \alpha \quad (\text{so that } \frac{\ell - \alpha}{\alpha + 2g - 1} = p)\ , \tag{3}$$

and p satisfies $k\binom{p+3}{3} > n$, say

$$p \stackrel{\text{def}}{=} \left\lfloor \left(\frac{6n}{k}\right)^{1/3} \right\rfloor . \tag{4}$$

Step 1: Find a **nonzero** trivariate polynomial $Q[Y, Z_1, Z_2]$ with coefficients in $\text{Ł}((\alpha-1)P_\infty)$ of **total degree** p, i.e., of the form

$$Q[Y, Z_1, Z_2] = \sum_{\substack{j, j_1, j_2 \\ j + j_1 + j_2 \leqslant p}} \Big(\sum_{r=1}^{k} \rho_{r,j,j_1,j_2} \phi_r \Big) Y^j Z_1^{j_1} Z_2^{j_2}\ ,$$

by finding the value of the unknowns $\rho_{r,j,j_1,j_2} \in \mathbb{F}_q$, such that for each $i = 1, 2, \ldots, n$, the constant term of the polynomial

$$Q^{(i)}[Y, Z_1, Z_2] \stackrel{\text{def}}{=} Q[Y + y_i, Z_1 + z_{i1}, Z_2 + z_{i2}]$$

vanishes at P_i.

Note that these conditions enforce a homogeneous linear system of equations over $\mathbb{F}_q$ in the unknowns ρ_{r,j,j_1,j_2}.

Step 2: Compute the polynomial $N \in \mathbb{F}_{q^\alpha}[Y, Z_1, Z_2]$ by evaluating each of the coefficients of Q (which are functions in $\text{Ł}((\alpha-1)P_\infty)$) at the place R.

Step 3: Compute the univariate polynomial $T \in \mathbb{F}_{q^\alpha}[Y]$ where $T[Y] \stackrel{\text{def}}{=} N[Y, Y^{s_1}, Y^{s_2}]$.

Step 4: Compute all the roots in $\mathbb{F}_{q^\alpha}$ of T. For each root $\gamma \in \mathbb{F}_{q^\alpha}$ of T, do the following:

- Compute the unique $f \in \text{Ł}((\alpha-1)P_\infty)$, if any, such that $f(R) = \gamma$ (this can also be accomplished by solving a linear system, with unknowns being the coefficients $a_1, \ldots, a_k$ of the basis elements $\phi_1, \ldots, \phi_k$ where $f = a_1\phi_1 + \cdots + a_k\phi_k$).
- If such an f exists, test if the encoding of f, $\mathsf{Enc}(f)$ that is defined in (2), agrees with the input triples on at least t locations. If so, output f.

5.3. **Runtime analysis.**

Lemma 5.2. *The above algorithm can be implemented to run in polynomial time, given an appropriate representation of the code, that consists of:*

(i) *The evaluation of the basis elements* $\phi_1, \ldots, \phi_k$ *of* $L((\alpha-1)P_\infty)$ *at the places* $P_1, \ldots, P_n$*, as well as at a place* R *of degree* α*.*
(ii) *The evaluation of* $\psi_1, \ldots, \psi_{2g}$ *at the places* $P_1, \ldots, P_n, R$*, where* $\phi_1, \ldots, \phi_k$*,* $\psi_1, \ldots, \psi_{2g}$ *forms a basis of* $L((\alpha+2g-1)P_\infty)$*.*

Proof. Our goal is to describe why the above information suffices for efficient decoding. With the information in (i), one can perform

- Step 1 using the values of ϕ_i's at $P_1, \ldots, P_n$ by solving a homogeneous linear system over $\mathbb{F}_q$;
- Step 2 using the values of the ϕ_i's at R; and
- the computation of f (if any) satisfying $f(R) = \gamma$ in Step 4, again using the values of the ϕ_i's at R.

Using the information in (ii) one can compute the map $I : \mathbb{F}_{q^\alpha} \to L((\alpha+2g-1)P_\infty)$ and thus compute $\mathsf{Enc}(f)$ in Step 4 and check which of the f's that are found must be output. The root-finding in Step 4 can be performed in deterministic $\mathrm{poly}(n, q, \alpha)$ time. Therefore, the overall runtime will be polynomial in the block length n. □

5.4. **Analysis of error-correction performance.** In this subsection, our goal is to prove the following concerning the performance of the above decoding algorithm.

Theorem 5.3. *For the choice of parameters* $p = \lfloor (6n/k)^{1/3} \rfloor$*,* $s_1 = p+1$ *and* $s_2 = p^2+p+1$*, the above decoding algorithm correctly finds all codewords* $c = \langle c_1, \ldots, c_n \rangle$ *of* C *which satisfy* $c_i = y_i + \beta_1 z_{i1} + \beta_2 z_{i2}$ *for at least* t *values of* $i \in \{1, 2, \ldots, n\}$*, for*

$$t = k + g + \left(6\Big(1 + \frac{3g-1}{k}\Big)\right)^{1/3} \cdot \Big((k+3g-1)^2 n\Big)^{1/3}. \tag{5}$$

The number of codewords the algorithm outputs in the worst-case is at most

$$p(p^2+p+1) \leqslant 3p^3 \leqslant 18n/k.$$

We will prove Theorem 5.3 by a sequence of lemmas.

Lemma 5.4. *For parameters* p, ℓ *defined in Step 0 of the algorithm, Step 1 of the algorithm finds a nonzero polynomial* $Q[Y, Z_1, Z_2]$ *of total degree* p *that satisfies the interpolation conditions of Section* 5.1*.*

Proof. Step 1 of the algorithm finds a polynomial $Q[Y, Z_1, Z_2]$ of total degree p whose coefficients lie in $L((\alpha-1)P_\infty)$. The coefficient of $Y^j Z_1^{j_1} Z_2^{j_2}$ is expressed using the unknowns ρ_{r,j,j_1,j_2} for $1 \leqslant r \leqslant k$. The total number of unknowns is thus k times the number of trivariate monomials of total degree at most p, which is $\binom{p+3}{3}$, and thus equals $k\binom{p+3}{3}$. The number of homogeneous linear conditions imposed on the unknowns is n, one for each place P_i. Therefore, if $k\binom{p+3}{3} > n$, the number of unknowns exceeds the number of constraints, and so a nonzero Q can be found.

It remains to prove that any Q that is found satisfies the following two conditions:

(a) for all $f, h_1, h_2 \in L((\alpha+2g-1)P_\infty)$, $Q(f, h_1, h_2) \in L((\ell-1)P_\infty)$, and
(b) for each $i = 1, 2, \ldots, n$, if f, h_1, h_2 evaluate to y_i, z_{i1}, z_{i2} respectively at P_i, then $Q(f, h_1, h_2)$ vanishes at P_i.

Condition (a) is immediate. Indeed, each monomial of Q has degree p and each coefficient of Q belongs to $L((\alpha-1)P_\infty)$. Therefore, for

$$f, h_1, h_2 \in L((\alpha+2g-1)P_\infty),$$

$Q(f, h_1, h_2)$ will have at most $(\alpha - 1) + p(\alpha + 2g - 1) = \ell - 1$ poles at P_∞, and no poles elsewhere.

For (b), we note the following:

$$\begin{aligned} \mathsf{ev}_{P_i}\big(Q(f, h_1, h_2)\big) &= \mathsf{ev}_{P_i}\big(Q^{(i)}(f - y_i, h_1 - z_{i1}, h_2 - z_{i2})\big) \\ &= \mathsf{ev}_{P_i}\big(Q^{(i)}(f - f(P_i), h_1 - h_1(P_i), h_2 - h_2(P_i))\big) \\ &= 0 \end{aligned}$$

where the last equality follows since by construction of Q, the constant term of $Q^{(i)}(Y, Z_1, Z_2)$ vanishes at P_i, and the functions $f - f(P_i)$, $h_1 - h_1(P_i)$ and $h_1 - h_2(P_i)$ all clearly vanish at P_i. □

Lemma 5.5. *If $Q \not\equiv 0$, then $N \in \mathbb{F}_{q^\alpha}[Y, Z_1, Z_2]$ obtained in Step 2 is a nonzero polynomial of total degree at most p. Moreover, if $Q(f, h_1, h_2) = 0$ for some functions $f, h_1, h_2 \in \mathcal{O}_R$, then $N(f(R), h_1(R), h_2(R)) = 0$.*

Proof. Q has total degree at most p, and hence so does N. Also, any nonzero coefficient of Q evaluates to a nonzero value at R since the map ev_R is injective by Lemma 4.1. Therefore, if $Q \not\equiv 0$, then N is a nonzero polynomial. Since the evaluation map $\mathsf{ev}_R : \mathcal{O}_R \to \mathbb{F}_{q^\alpha}$ is a homomorphism, $N(f(R), h_1(R), h_2(R))$ equals the evaluation of $Q(f, h_1, h_2)$ at R, and so must equal 0 if $Q(f, h_1, h_2) = 0$. □

Lemma 5.6. *If $p \geqslant 1$, $s_1 > p$, $s_2 > s_1 p$, and $N[Y, Z_1, Z_2]$ is a nonzero polynomial of total degree at most p, then the polynomial $T[Y] = N[Y, Y^{s_1}, Y^{s_2}]$ is a nonzero polynomial of degree at most $s_2 p$.*

Proof. The claim about the degree of T is obvious, so we just need to show that T is nonzero. Define the polynomial $S[Y, Z_2] \stackrel{\text{def}}{=} N[Y, Y^{s_1}, Z_2]$. Now $S \equiv 0$ iff $Z_1 - Y^{s_1}$ divides $N[Y, Z_1, Z_2]$. But this is impossible since the total degree of N is at most $p < s_1$. Therefore, S is a nonzero polynomial of total degree at most $s_1 p$. Now, the polynomials T, S are related by $T[Y] = S[Y, Y^{s_2}]$. Therefore, $T \equiv 0$ iff $Z_2 - Y^{s_2}$ divides $S[Y, Z_2]$. Again this is impossible since the degree of S is at most $s_1 p < s_2$. We conclude that T must be a nonzero polynomial. □

Combining the above lemmas, it is easy to conclude that:

Lemma 5.7. *For ℓ, p defined as in Step 0, and the choices $s_1 = p + 1$ and $s_2 = p(p+1) + 1$, for every $f \in L((\alpha - 1)P_\infty)$, the following holds: If $\mathsf{Enc}(f)$ agrees with the input word on ℓ or more places, then $f(R)$ is a root of T, and thus f will be found and output in Step 4 of the algorithm. Moreover, the algorithm will output at most $p^3 + p^2 + p$ such functions f.*

Proof. By Lemma 5.4 and Lemma 5.1, each $f \in \mathrm{L}((\alpha - 1)P_\infty)$ for which $\mathsf{Enc}(f)$ has agreement $\geqslant \ell$ with the input word, satisfies $Q(f, h_1, h_2) = 0$, where $h_1 = I[f(R)^{s_1}]$ and $h_2 = I[f(R)^{s_2}]$. By Lemma 5.5, $N(f(R), h_1(R), h_2(R)) = 0$. Hence

$$T(f(R)) = N(f(R), f(R)^{s_1}, f(R)^{s_2}) = N(f(R), h_1(R), h_2(R)) = 0 \ .$$

Thus $f(R)$ is a root of T. By Lemma 5.6, T is a nonzero polynomial of degree at most $s_2 p$. It follows that the number of solutions f output by the algorithm is at most $s_2 p = p^3 + p^2 + p$. □

Proof of Theorem 5.3. Theorem 5.3 follows immediately from Lemma 5.7 and the choice of ℓ in (3): $\ell = \alpha + (\alpha + 2g - 1)p$ where $p = (6n/k)^{1/3}$ and $\alpha = k + g$. □

For small rates, the result of Theorem 5.3 improves over the list decoding algorithm for AG codes in [6] which corrects up to $n - \sqrt{(k+g-1)n}$ errors.

5.5. **Consequences.** So far our construction applies to any function field. We conclude this section by stating the following corollary to Theorem 5.3 obtained by plugging in function fields with the best possible ratio of g/n. Specifically, for q a square, we will use a sequence of function fields with increasing genus for which g/n is at least $\frac{1}{\sqrt{q}-1}$ [15, 1]. (The claim about the polynomial time constructibility of the codes follows from Section 7.)

Theorem 5.8. *For q a square prime power and every $\mathcal{R}$, $\frac{1}{\sqrt{q}-1} < 3\mathcal{R} < 1 - \frac{1}{\sqrt{q}-1}$, there is a family of codes over alphabet size q^3 of rate $\mathcal{R}$, relative distance at least $1 - 3\mathcal{R} - \frac{1}{\sqrt{q}-1}$, and which is list decodable up to a fraction $\left(1 - 3\mathcal{R} - \frac{1}{\sqrt{q}-1} - 6\left(\mathcal{R} + \frac{1}{\sqrt{q}-1}\right)^{2/3}\right)$ of errors[6] using lists of size at most $6/\mathcal{R}$. Furthermore, there is a natural representation of the codes, computable in expected polynomial time, for which the encoding as well as list decoding up to this radius can be performed in polynomial time.*

For decoding up to a fraction of errors approaching 1, we get the following corollary. (We say a code of block length n is (ρ, L)-list decodable if for every received word there are at most L codewords within distance ρn from it.)

Corollary 5.9. *For all small enough $\varepsilon > 0$, there is a family of Q-ary codes for $Q = O(1/\varepsilon^9)$ which has rate $\Omega(\varepsilon^{3/2})$ and which is $(1-\varepsilon, O(1/\varepsilon^{3/2}))$-list decodable. Furthermore, the codes have a representation, computable in expected polynomial time, that permits polynomial time encoding and list decoding up to radius $(1-\varepsilon)$.*

The above corollary can be contrasted with regular AG codes that are list decodable up to radius $(1-\varepsilon)$ using the algorithm in [6]. Those codes had a worse rate of $\Theta(\varepsilon^2)$, but their alphabet size was $O(1/\varepsilon^4)$. The above gives the first codes with a rate better than $\Omega(\varepsilon^2)$ for list decoding up to a fraction $(1-\varepsilon)$ of errors over an alphabet of size polynomial in $1/\varepsilon$.

5.6. **Extension to higher order correlations.** We can modify the basic construction of Section 4 by using $m \geqslant 4$ correlated functions $f, h_1, h_2, \ldots, h_{m-1}$ to perform the encoding. The function $f \in \text{Ł}((\alpha-1)P_\infty)$ will be the message, and the functions $h_i \in \text{Ł}((\alpha+2g-1)P_\infty)$ will be defined by $h_i = I[f(R)^{s_i}]$ for suitable choices of $s_1, s_2, \ldots, s_{m-1}$. The rate of the code is $k/(mn)$ and its distance at least $n-k-g+1$.

For the decoding, in order to find a nonzero interpolation polynomial Q, the parameter p in (4) must now satisfy $k\binom{p+m}{m} > n$ since the number of monomials in a total degree p m-variate polynomial equals $\binom{p+m}{m}$. The above condition is satisfied for the choice $p = \lfloor (m!n/k)^{1/m} \rfloor$. The choice of ℓ remains the same as in (3). For the choice $s_1 = p+1$ and $s_i = ps_{i-1} + 1$ for $i \leqslant 2 \leqslant m-1$, a decoding algorithm similar to the one in Section 5.2, finds all codewords with agreement at

[6]When the stated fraction of errors is nonpositive, the stated bound becomes trivial. So the result is meaningful only for small rates $\mathcal{R}$.

least t with any input word, where

$$t = k + g + \left(m!\Big(1 + \frac{3g-1}{k}\Big)\right)^{\frac{1}{m}} \cdot \Big((k+3g-1)^{m-1}n\Big)^{\frac{1}{m}} . \tag{6}$$

The size of list output will be at most $s_{m-1}p = \sum_{i=1}^{m} p^i \leqslant mp^m$. Using the above with the function fields of best possible g/n ratio, we get the following generalizations of Theorem 5.8 and Corollary 5.9.

Theorem 5.10 (Main)**.** *For q a square prime power, an integer $m \geqslant 3$, and every $\mathcal{R}$ satisfying $\frac{1}{\sqrt{q}-1} < m\mathcal{R} < 1 - \frac{1}{\sqrt{q}-1}$, there is a family of codes over alphabet size q^m with rate $\mathcal{R}$, relative distance at least $1 - m\mathcal{R} - \frac{1}{\sqrt{q}-1}$, and which is list decodable up to a fraction $\left(1 - m\mathcal{R} - \frac{1}{\sqrt{q}-1} - (4m!)^{1/m}\Big(m\mathcal{R} + \frac{3}{\sqrt{q}-1}\Big)^{1-1/m}\right)$ of errors using lists of size at most $m!/\mathcal{R}$. Moreover, there is a natural representation of the codes, computable in expected polynomial time, for which the encoding as well as list decoding up to this radius can be performed in polynomial time.*

For decoding up to a fraction of errors approaching 1, we get the following corollary.

Corollary 5.11. *For all $\varepsilon > 0$ and all integers $m \geqslant 3$, there is a family of Q-ary codes for $Q = O((m/\varepsilon)^{\frac{2m^2}{m-1}})$ which has rate $\Omega(\frac{1}{m^2} \cdot \varepsilon^{m/(m-1)})$ and which is $(1 - \varepsilon, O(m^2 m!(1/\varepsilon)^{m/(m-1)}))$-list decodable. Moreover, the codes have a representation, computable in expected polynomial time, that permits polynomial time list decoding up to radius $(1 - \varepsilon)$.*

The above gives the first codes with rate better than $\Omega(\varepsilon^2)$ for list decoding up to a fraction $(1 - \varepsilon)$ of errors over an alphabet of size polynomial in $1/\varepsilon$. To maximize the rate as a function of ε (which we think of as a small constant), we can pick $m = \Theta(\log(1/\varepsilon))$ in the above corollary.

Corollary 5.12. *For all $\varepsilon > 0$, there is a family of Q-ary codes for*

$$Q = (1/\varepsilon)^{O(\log(1/\varepsilon))}$$

which has rate $\Omega(\varepsilon/\log^2(1/\varepsilon))$ and which is $(1-\varepsilon, (1/\varepsilon)^{O(\log\log(1/\varepsilon))})$-list decodable. Moreover, the codes have a natural representation, computable in expected polynomial time, that permits polynomial time encoding as well as polynomial time list decoding up to radius $(1 - \varepsilon)$.

6. A second decoding algorithm

We now describe our second decoding algorithm, which uses the second approach described in Section 2 to address the problem of all coefficients of the interpolated polynomial vanishing at R. We consider only the case of two correlated functions to keep the exposition simple. The idea can be extended to three or more correlated functions in a straightforward way. Note that for a technical reason, we needed three or more correlated functions for the algorithm in Section 5 to give an improvement over AG codes. Here no such technicalities arise. We therefore first restate the problem in its two correlated functions version.

Input: Pairs $(y_i, z_i) \in \mathbb{F}_q^2$, for $i = 1, 2, \ldots, n$.

Output: All functions $f \in \text{Ł}((\alpha-1)P_\infty)$ for which the tuple of functions $(f,\ h = I[f(R)^{s_1}])$ satisfies $f(P_i) = y_i$, $h(P_i) = z_i$ for at least t values of $i \in \{1, 2, \ldots, n\}$.

6.1. **High level idea behind the algorithm.** We follow the same interpolation based decoding idea from Section 5. However, in a major departure here we allow higher multiplicities as in [6]. In the interpolation step, we try to fit the data points $\{(P_i, y_i, z_i)\}_{i=1}^n$ by a polynomial $Q(Y, Z) \in K[Y, Z]$ with the following properties (for suitable parameter choices for ℓ, r):

(1) Q is nonzero.
(2) For all $f, h \in \text{Ł}((\alpha + 2g - 1)P_\infty), Q(f, h) \in L(\ell P_\infty)$.
(3) For every $i \in [n]$, for all $f, h \in \text{Ł}((\alpha + 2g - 1)P_\infty)$ which satisfy $f(P_i) = y_i$, $h(P_i) = z_i$, $Q(f, h)$ has a zero of multiplicity r at P_i i.e., $v_{P_i}(Q(f, h)) \geqslant r$.

The following lemma is analogous to Lemma 5.1.

Lemma 6.1. *Let Q satisfy above conditions. Further, let $f, h \in \text{Ł}((\alpha+2g-1)P_\infty)$ satisfy $f(P_i) = y_i$ and $h(P_i) = z_i$ for at least t values of $i \in [n]$, and $rt > \ell$; then $Q(f, h) \equiv 0$.*

Proof. By Property 2 of the interpolated polynomial Q, $v_{P_\infty}(Q(f, h)) \geqslant -\ell$. However, at least on t points it holds that $f(P_i) = y_i$ and $h(P_i) = z_i$. Therefore $\sum_{i=1}^n v_{P_i}(Q(f, h)) \geqslant r \cdot t > \ell$. Hence $Q(f, h)$ must be identically zero. □

As before, we then reduce the polynomial Q modulo the place R, by evaluating each of its coefficients at R, to obtain a polynomial $N(Y, Z) \in \mathbb{F}_{q^\alpha}[Y, Z]$. At this step, as mentioned earlier, we would be stuck if N is the zero polynomial. To solve this problem, we exploit the following facts. Since the degree of R is large, there exists a function, say ν, that has a pole of order one at R and has no other pole. (To see this, observe that if $\deg(R)$ is large, then by Riemann-Roch Ł(R) is nonempty. Hence there exists a rational function that has pole of order one at R and nowhere else.) Also, each coefficient of Q has at most $w = \lfloor \ell/\alpha \rfloor$ zeroes at R. Therefore, if $N \equiv 0$, there must exist a minimum c, $1 \leqslant c \leqslant w$, such that $\tilde{Q} = \nu^c Q$ has a coefficient that does not vanish at R. Clearly if $Q(f, h) \equiv 0$, then $\tilde{Q}(f, h) \equiv 0$ as well. Therefore, if we can find $\tilde{Q}$ and reduce it modulo R, we will get a nonzero polynomial N such that $N(f(R), h(R)) = 0$. Further setting $T(Y) = N(Y, Y^{s_1})$ as before, we would have $T(f(R)) = 0$, and the task of finding all message functions f reduces to finding all the roots of the univariate polynomial T. Again, we need to make sure that the reduction does not produce the zero polynomial. This is ensured as in the algorithm before by choosing a suitably large s_1.

The whole issue, therefore, is how to find $\tilde{Q} = \nu^c Q$ with the stated property. The coefficients of $\nu^c Q$ all belong to the linear space Ł$(\ell P_\infty + wR)$. It turns out that we can find c, and evaluate all coefficients of $\tilde{Q}$ at R using linear algebra in this linear space, assuming preprocessed information about the evaluations of functions in a suitable basis of Ł$(\ell P_\infty + wR)$ at R. This yields a polynomial time decoding algorithm given access to a polynomial amount of preprocessed information concerning the code — details follow.

6.2. **Formal description of the decoding algorithm using multiplicities.** We now formally specify our second decoding algorithm (that was outlined in Section 6.1). Recall that the input to the decoding algorithm is a set of n pairs $(y_i, z_i) \in \mathbb{F}_q^2$, and the algorithm should find all functions $f \in \text{Ł}((\alpha-1)P_\infty)$ such that $(f(P_i), h(P_i)) = (y_i, z_i)$ for at least t values of $i \in \{1, 2, \ldots, n\}$, where $h = I[f(R)^{s_1}]$ is the function in $\text{Ł}((\alpha + 2g - 1)P_\infty)$ correlated with f. The agreement parameter t will come out from the analysis. In what follows, $\phi_1, \phi_2, \ldots, \phi_k$ denotes a basis of $\text{Ł}((\alpha - 1)P_\infty)$ (recall that this is a k-dimensional vector space over $\mathbb{F}_q$).

Our algorithm will run in polynomial time assuming some polynomial amount of preprocessed information about the code, as described below (ℓ is a parameter as defined in (9), $b = \ell - g + 1$, and $w \stackrel{\text{def}}{=} \lfloor \frac{\ell}{\alpha} \rfloor$):

(1) The evaluation at the places $P_1, P_2, \ldots, P_n, R$ of a basis $\mathcal{B}$ of $\text{Ł}(\ell P_\infty)$ with increasing pole orders at P_∞, i.e., functions $\psi_1, \ldots, \psi_b$ such that
   - $\forall j \in [b]\ v_{P_\infty}(\psi_j) \geqslant 1 - g - j$ and
   - $\forall j \in [b-1]\ v_{P_\infty}(\psi_j) > v_{P_\infty}(\psi_{j+1})$.

   The first $\alpha + g$ of these functions $\psi_1, \ldots, \psi_{\alpha+g}$ form a basis of
   $$\text{Ł}((\alpha + 2g - 1)P_\infty)$$
   and their evaluations are all that is needed for the encoding. For the decoding, we also need evaluations of the rest of the basis functions, as well as additional information described next.

(2) The coefficients for a change of basis that expresses $\mathcal{B}$ in terms of a *zero-increasing* basis of $\text{Ł}(\ell P_\infty)$ w.r.t. place P_i for each $i = 1, 2, \ldots, n$. Formally, coefficients $\gamma_{i,j,h} \in \mathbb{F}_q$ for $i = 1, 2, \ldots, n$ and $1 \leqslant j, h \leqslant b$ such that there exist $\theta_1^{(i)}, \ldots, \theta_b^{(i)} \in K^*$ with $v_{P_i}(\theta_j) \geqslant j - 1$ satisfying
   $$\psi_j = \sum_{h=1}^{b} \gamma_{i,j,h} \theta_h^{(i)}. \tag{7}$$
   Lemma 6.5 asserts the existence of such basis $\theta_1^{(i)}, \ldots, \theta_b^{(i)}$ for each i.

(3) The following information about a basis $\mathcal{B}' = \{\psi_1, \psi_2, \ldots, \psi_b\} \cup \{\zeta_{ij} \mid 1 \leqslant i \leqslant \alpha, 1 \leqslant j \leqslant w\}$ for $\text{Ł}(\ell P_\infty + wR)$ (this basis extends the basis $\mathcal{B}$ which we had for $\text{Ł}(\ell P_\infty)$). For some $\nu \in \text{Ł}(R)$ that has one pole at R and no poles at any other place (such a ν exists by the Riemann-Roch theorem if $\deg(R) = \alpha > g$), assume we know the expansion in the basis $\mathcal{B}'$ (i.e., the $b + w\alpha$ coefficients in $\mathbb{F}_q$) for each of the functions $\psi_i \nu^c$ for $1 \leqslant i \leqslant b$ and $1 \leqslant c \leqslant w$. (Note that each such function belongs to $\text{Ł}(\ell P_\infty + wR)$ since $\psi_i \in \text{Ł}(\ell P_\infty)$ and $\nu^c \in \text{Ł}(cR) \subseteq \text{Ł}(wR)$.)

Armed with this preprocessed representation of the code, we are now ready to describe the algorithm in detail.

Step 0: Compute integer parameters r, ℓ where
$$r \stackrel{\text{def}}{=} \left\lceil \frac{\alpha + 3g + 2\sqrt[3]{n(\alpha + 2g - 1)^2}}{t - \sqrt[3]{n(\alpha + 2g - 1)^2}} \right\rceil \tag{8}$$
and
$$\ell \stackrel{\text{def}}{=} rt - 1. \tag{9}$$
Set $b \stackrel{\text{def}}{=} \ell - g + 1$ and $\sigma \stackrel{\text{def}}{=} \lfloor \frac{\ell - g}{\alpha + 2g - 1} \rfloor$.

Step 1: (Interpolation step) Find a **nonzero** bivariate polynomial $Q[Y, Z]$ with coefficients in $\text{Ł}(\ell P_\infty)$ of **total degree** σ, i.e., of the form

$$Q[Y,Z] = \sum_{\substack{j_2+j_3 \leqslant \sigma \\ j_2,j_3 \geqslant 0}} \sum_{j_1=1}^{b-(\alpha+2g-1)(j_2+j_3)} \rho_{j_1,j_2,j_3} \psi_{j_1} Y^{j_2} Z^{j_3}, \tag{10}$$

by finding the value of the unknowns $\rho_{j_1,j_2,j_3} \in \mathbb{F}_q$, such that for each $i \in [n]$, the polynomial $Q^{(i)}[Y, Z] \stackrel{\text{def}}{=} Q[Y + y_i, Z + z_i]$ vanishes at $(P_i, 0, 0)$ with multiplicity r.

(Using the representation of each basis function ψ_{j_i} in the zero-increasing basis for P_i as assumed in (7), we can find such a Q by solving a homogeneous linear system of equations over $\mathbb{F}_q$ in the unknowns ρ_{j_1,j_2,j_3}. See the proof of Lemma 6.4 for details.)

Step 2: Compute the polynomial $N \in \mathbb{F}_{q^\alpha}[Y, Z]$ by evaluating each of the coefficients of Q (which are functions in $\text{Ł}(\ell P_\infty)$ expressed in the basis $\{\psi_1, \ldots, \psi_b\}$) at the place R. If $N \not\equiv 0$, proceed to Step 5.

Step 3: (Dealing with $N \equiv 0$) Define the set of nonzero coefficients of Q

$$E \stackrel{\text{def}}{=} \{\eta_{ij} \mid Q[Y,Z] = \sum_{ij} \eta_{ij} Y^i Z^j, 0 \neq \eta_{ij} \in \text{Ł}(\ell P_\infty)\}.$$

Now compute

$$c^* \stackrel{\text{def}}{=} \min_{\eta_{ij} \in E} \max_{c \in \{0,1,\ldots,w\}} \{c \mid \nu^c \eta_{ij} \text{ when expanded in basis } \mathcal{B}' \text{ has zero coefficients for all } \zeta_{ij}\}.$$

(Note that having zero coefficients for all ζ_{ij} implies that the function has no pole at R. Also, $c^* \geqslant 1$ since each η_{ij} has a zero at R and so can be multiplied by a positive power of ν and still not have a pole at R.) The above computation of c^* can be done by simply checking when the expansion of $\eta_{ij}\nu^c$ in basis $\mathcal{B}'$ has zero coefficients for all the ζ_{ij}'s.

Step 4: Compute the polynomial $N[Y, Z] \stackrel{\text{def}}{=} (\nu^{c^*} Q[Y, Z])(R) \in \mathbb{F}_{q^\alpha}[Y, Z]$ using the given evaluations of each ψ_i at R (by the definition of c^* all coefficients of $\nu^{c^*} Q$ are regular at R). Note that $N \neq 0$ by the definition of c^*.

Step 5: Compute the univariate polynomial $T \in \mathbb{F}_{q^\alpha}[Y]$ where

$$T[Y] \stackrel{\text{def}}{=} N[Y, Y^{s_1}].$$

Step 6: Compute all the roots in $\mathbb{F}_{q^\alpha}$ of T. For each root $\gamma \in \mathbb{F}_{q^\alpha}$ of T, do the following:

- Compute the unique $f \in \text{Ł}((\alpha-1)P_\infty)$, if any, such that $f(R) = \gamma$ (this can also be accomplished by solving a linear system, with unknowns being the coefficients $a_1, \ldots, a_k$ of the basis elements $\phi_1, \ldots, \phi_k$ where $f = a_1\phi_1 + \cdots + a_k\phi_k$).
- If such an f exists, test if the encoding of $(f, I[f(R)^{s_1}])$ agrees with the input tuples on at least t locations. If so, output f.

6.3. **Runtime analysis of the algorithm.** The above algorithm can be implemented to run in polynomial time using the preprocessed information assumed in the previous section along with the following assumed representation.

Lemma 6.2. *The above algorithm can be implemented to run in polynomial time, given an appropriate representation of the code, that consists of:*

(i) *The evaluation of the basis elements of* $L(\ell P_\infty)$ *with increasing pole orders (i.e., functions* $\psi_1, \ldots, \psi_b$*) at the places* $P_1, \ldots, P_n$*, as well as at a high degree place* R *of degree* α*.*

(ii) *The coefficients for a change of basis that expresses* $\mathcal{B}$ *in terms of zero-increasing basis of* $L(\ell P_\infty)$ *w.r.t. place* P_i *for each* $i = 1, \ldots, n$*.*

(iii) *An explicit basis* $\mathcal{B}' = \{\psi_1, \ldots, \psi_b\} \cup \{\zeta_{ij} | 1 \leqslant i \leqslant \alpha, 1 \leqslant j \leqslant w\}$ *of* $L(\ell P_\infty + wR)$ *as well as expansion of* $\psi_i \nu^c$ *for* $1 \leqslant i \leqslant b$*,* $1 \leqslant c \leqslant w$ *in the basis* $\mathcal{B}'$*.*

Proof. Clearly the encoding is efficient given the information in (i). Thus we need to show that the above information suffices for an efficient decoding.

- Step 1 is essentially solving a homogeneous linear system over $\mathbb{F}_q$ using the representation (given in (ii)) of each basis functions in the zero-increasing basis for P_i.
- Step 2 uses the given evaluations (given in (i)) of ψ_i's at the high degree place R.
- Step 3 uses the information given in (iii) and hence is efficient.
- Step 4 uses the information given in (i), i.e., it uses evaluations of the basis functions in $L(\ell P_\infty)$ at R.
- Step 5 is a simple substitution.
- Step 6 can be solved efficiently by a root finding algorithm that runs in deterministic $\mathrm{poly}(q, n, \alpha)$ time, followed by elementary linear algebraic operations.

Thus the overall runtime will be polynomial in the block length n. □

6.4. **Analysis of error-correction performance.**

Theorem 6.3. *For the choice of* $s_1 = \sigma + 1$*, the decoding algorithm of Section* 6.2 *correctly finds all the codewords* $c = \langle c_1, \ldots, c_n \rangle$ *of* C *which satisfy* $c_i = y_i + \beta z_i$ *for at least* t *values of* $i \in [n]$*, for* $t > \sqrt[3]{n(k+3g-1)^2}$*. Moreover, for some* $c > 1$*, if* $t \geqslant c\sqrt[3]{n(k+3g-1)^2}$*, then the list output by the algorithm has size at most* $O((\frac{c}{c-1})^2 (n/k)^{2/3})$*.*

We will prove Theorem 6.3 by a sequence of lemmas.

Lemma 6.4. *For parameters* r, ℓ *defined in Step 0 of the algorithm, Step* 1 *finds a* nonzero *polynomial* $Q[Y, Z]$ *satisfying the following two conditions:*

(1) *For all* $f, h \in L((\alpha + 2g - 1)P_\infty), Q(f, h) \in L(\ell P_\infty)$*.*

(2) *For every* $i \in [n]$*, for all* $f, h \in L((\alpha + 2g - 1)P_\infty)$ *which satisfy* $f(P_i) = y_i$*,* $h(P_i) = z_i$*,* $Q(f, h)$ *has a zero of multiplicity* r *at* P_i *i.e.,* $v_{P_i}(Q(f, h)) \geqslant r$*.*

Proof. First note that by the way Q is expressed in equation (10), for any $f, h \in L((\alpha + 2g - 1)P_\infty)$ it holds that

$$\begin{aligned} v_{P_\infty}(Q(f,h)) &\geqslant (v_{P_\infty}(\psi_{j_1}) + j_2 \cdot v_{P_\infty}(f) + j_3 \cdot v_{P_\infty}(h)) \\ &\geqslant 1 - g - j_1 - (\alpha + 2g - 1)(j_2 + j_3) \geqslant -\ell \ . \end{aligned}$$

Also clearly $Q(f,h)$ has no poles outside P_∞. Therefore, the first of the two required conditions is satisfied. To get the second condition, we begin with a lemma from [6], that allows for a change into a basis with an increasing number of zeroes at any desired place P_i.

Lemma 6.5. *Given functions $\psi_1,\ldots,\psi_b \in L(\ell P_\infty)$ of distinct orders at P_∞ satisfying $v_{P_\infty}(\psi_j) \geqslant 1-g-j$ and a rational point $P_i \neq P_\infty$, there exists $\theta_1^{(i)},\ldots,\theta_b^{(i)} \in K^*$ with $v_{P_i}(\theta_j) \geqslant j-1$ and $\gamma_{i,j,h} \in \mathbb{F}_q$ for all j,h with $1 \leqslant j,h \leqslant b$ such that*

$$\psi_j = \sum_{h=1}^{b} \gamma_{i,j,h}\theta_h^{(i)}. \tag{11}$$

Using the above we will express the condition that "$Q^{(i)}[Y,Z] \stackrel{\text{def}}{=} Q[Y+y_i,Z+z_i]$ vanishes at $(P_i,0,0)$ with multiplicity r" as a collection of homogeneous linear equations in the unknowns ρ_{j_1,j_2,j_3} describing Q. Expressing $Q[Y,Z]$ in the basis $\theta_h^{(i)}$ for $1 \leqslant h \leqslant b$, we get

$$Q[Y,Z] = \sum_{\substack{j_2+j_3\leqslant\sigma\\ j_2,j_3\geqslant 0}} \sum_{j_1=1}^{b-(\alpha+2g-1)(j_2+j_3)} \sum_{h=1}^{b} \rho_{j_1,j_2,j_3}\gamma_{i,j_1,h}\theta_h^{(i)} Y^{j_2}Z^{j_3}. \tag{12}$$

The shifting to y_i, z_i is achieved by defining $Q^{(i)}[Y,Z] \stackrel{\text{def}}{=} Q[Y+y_i,Z+z_i]$. Note that the terms in $Q^{(i)}[Y,Z](P_i)$ that are divisible by Y^uZ^v contribute $(u+v)$ towards the multiplicity of $(P_i,0,0)$ as a zero of $Q^{(i)}$, or equivalently, the multiplicity of (P_i,y_i,z_i) as a zero of Q. Then

$$Q^{(i)}[Y,Z] = \sum_{\substack{j_4+j_5\leqslant\sigma\\ j_4,j_5\geqslant 0}} \sum_{h=1}^{b} w_{h,j_4,j_5}^{(i)}\theta_h^{(i)} Y^{j_4}Z^{j_5}, \tag{13}$$

where

$$w_{h,j_4,j_5}^{(i)} \stackrel{\text{def}}{=} \sum_{\substack{j_2=j_4\\ j_3=j_5}}^{j_2+j_3\leqslant\sigma} \sum_{j_1=1}^{b-(\alpha+2g-1)(j_2+j_3)} \binom{j_2}{j_4}\binom{j_3}{j_5} y_i^{j_2-j_4} z_i^{j_3-j_5} \rho_{j_1,j_2,j_3}\gamma_{i,j_1,h}. \tag{14}$$

Thus we want $w_{h,j_4,j_5}^{(i)} = 0$ for all $h \geqslant 1, j_4 \geqslant 0, j_5 \geqslant 0$ such that $j_4+j_5+(h-1) \leqslant (r-1)$, which is a collection of $\binom{r+2}{3}$ linear constraints. For all the n places $P_1,\ldots,P_n$, we have in total $n\binom{r+2}{3}$ homogeneous linear constraints. Thus, we can find the polynomial Q by solving a linear system.

We now prove that the second condition is satisfied:

Lemma 6.6. *Suppose we find a polynomial Q satisfying $w_{h,j_4,j_5}^{(i)} = 0$ for all $h \geqslant 1, j_4 \geqslant 0, j_5 \geqslant 0$ such that $j_4+j_5+(h-1) \leqslant (r-1)$, and all i. Then, if $f,h \in L((\alpha+2g-1)P_\infty)$ satisfy $f(P_i) = y_i$ and $h(P_i) = z_i$, then $Q(f,h)$ has a zero of multiplicity r at P_i i.e., $v_{P_i}(Q(f,h)) \geqslant r$.*

Proof. We have $Q(f,h) = Q^{(i)}(f-y_i,h-z_i) = Q^{(i)}(f-f(P_i),h-h(P_i))$, so that

$$Q(f,h) = \sum_{j_4+j_5\leqslant\sigma;j_4,j_5\geqslant 0} \sum_{h=1}^{b} w_{h,j_4,j_5}^{(i)}\theta_h^{(i)} (f-f(P_i))^{j_4}(h-h(P_i))^{j_5}\ .$$

Note that $f(P_i), h(P_i) \in \mathbb{F}_q$, so the above is a well-defined function in K. Since $w^{(i)}_{h,j_4,j_5} = 0$ for $j_4 + j_5 + (h-1) < r$, $v_{P_i}(\theta^{(i)}_h) \geqslant (h-1)$, $v_{P_i}((f - f(P_i))^{j_4}) \geqslant j_4$ and $v_{P_i}((h - h(P_i))^{j_5}) \geqslant j_5$, we obtain $v_{P_i}(Q(f,h)) \geqslant r$, as claimed. □

We have thus shown that the polynomial satisfies both the conditions required in the statement of Lemma 6.4. It remains to argue that the polynomial Q is in fact nonzero. Recall that the total number of linear constraints is $n \cdot r(r+1)(r+2)/6$. On the other hand the number of unknowns ρ_{j_1,j_2,j_3} equals the number of triples (j_1, j_2, j_3) such that $j_2, j_3 \geqslant 0$, $j_2 + j_3 \leqslant \sigma$, and $1 \leqslant j_i \leqslant b - (\alpha + 2g - 1)(j_2 + j_3)$. This number is easily seen to be at least $(\alpha+2g-1)\sigma(\sigma+1)(\sigma+2)/6$. Recall that a system of homogeneous equations always has a nonzero solution when the number of unknowns is larger than the number of constraints. Thus in order to ensure that a nonzero $Q[Y, Z]$ exists, we only need to ensure that

$$\frac{(\alpha + 2g - 1)\sigma(\sigma+1)(\sigma+2)}{6} \geqslant n \cdot \frac{r(r+1)(r+2)}{6} + 1. \tag{15}$$

The L.H.S is at least $(\alpha + 2g - 1)\sigma^3/6$ and $\sigma \geqslant \frac{\ell - g}{\alpha + 2g - 1} - 1$, so the above condition is met if we set $\ell \stackrel{\text{def}}{=} rt - 1$ and choose r so that it holds that

$$\frac{(\ell - \alpha - 3g + 1)^3}{(\alpha + 2g - 1)^2} \geqslant n(r+2)^3,$$

$$\text{i.e.,} \quad (rt - \alpha - 3g) \geqslant \sqrt[3]{n(\alpha + 2g - 1)^2} \cdot (r + 2).$$

To satisfy the above we can pick

$$r = \left\lceil \frac{\alpha + 3g + 2\sqrt[3]{n(\alpha + 2g - 1)^2}}{t - \sqrt[3]{n(\alpha + 2g - 1)^2}} \right\rceil, \tag{16}$$

which is exactly the choice made in Step 0 of the algorithm. Thus the choice of r, ℓ ensures a nonzero $Q[Y, Z]$. This proves that the interpolation step finds a polynomial $Q[Y, Z]$ satisfying all the required conditions. □

Lemma 6.7. *If $Q \not\equiv 0$, then $N[Y, Z] \in \mathbb{F}_{q^\alpha}[Y, Z]$ obtained in Step 4 is a nonzero polynomial of total degree at most σ. Moreover, if $Q(f, h) = 0$, for some functions $f, h \in \mathcal{O}_R$, then $N(f(R), h(R)) = 0$.*

Proof. First note that if $Q[Y, Z](R)$ is a zero polynomial, then $\forall \eta_{ij} \in E, v_R(\eta_{ij}) > 0$. Let c be the minimum order corresponding to some $\eta_{i^*j^*}$. Then it certainly holds that $v_R(\nu^c \eta_{i^*j^*}) = 0$ and $\forall v_R(\nu^c \eta_{ij}) \geqslant 0$. Hence, $(\nu^c Q[Y, Z])(R) \not\equiv 0$. Note that this c is exactly the c^* chosen in Step 3. Further note that since each $\eta_{ij} \in \text{Ł}(\ell P_\infty)$, hence $c^* \leqslant \lfloor \frac{\ell}{\deg R} \rfloor = \lfloor \frac{\ell}{\alpha} \rfloor$. Thus $N[Y, Z] \not\equiv 0$. The evaluation map $\mathsf{ev}_R : \mathcal{O}_R \to \mathbb{F}_{q^\alpha}$ is a homomorphism, so $N(f(R), h(R))$ equals the evaluation of $\nu^{c^*} Q(f, h)$ at R. Since $Q(f, h) = 0$ by assumption, we have $N(f(R), h(R)) = 0$. □

Lemma 6.8. *If $\sigma \geqslant 1$, $s_1 = \sigma + 1 > \sigma$ and $N[Y, Z]$ is a nonzero polynomial of total degree at most σ, then the polynomial $T[Y] = N[Y, Y^{s_1}]$ is a nonzero polynomial of degree at most $s_1\sigma$.*

Proof. The claim on the degree of T is straightforward. Therefore, we show that it is nonzero. Assume otherwise. Then $(Z - Y^{s_1}) | N[Y, Z]$, which is impossible since the degree of $N[Y, Z] < s_1$. □

Proof of Theorem 6.3. We first prove correctness, i.e., all messages f that should be output are indeed output by the algorithm. Let $f, h \in \mathrm{L}((\alpha+2g-1)P_\infty)$ be such that $h(R) = f(R)^{s_1}$ and $f(P_i) = y_i, h(P_i) = z_i$ for at least t points. Combining Lemma 6.1 and Lemma 6.4, we have that the polynomial Q found in Step 1 is nonzero and satisfies $Q(f,h) = 0$. Using Lemma 6.7, we have $f(R), h(R)$ satisfy $N(f(R), h(R)) = 0$ for the nonzero polynomial N. Since $s_1 > \sigma$, T is a nonzero polynomial. Finally, since $h(R) = f(R)^{s_1}$, we have $T(f(R)) = N(f(R), f(R)^{s_1}) = N(f(R), h(R)) = 0$. Hence $f(R)$ appears as a root in Step 6. We conclude that f appears as an output of the algorithm.

Regarding the claim about the list size, the size of the list output is at most the degree T which is at most $s_1\sigma = O(\sigma^2) = O(\ell^2/\alpha^2) = O((rt/\alpha)^2)$. With the choice of r as in (16), when $t \geqslant c\sqrt[3]{n(k+3g-1)^2}$, this bound is at most $O((\frac{c}{c-1})^2(n/k)^{2/3})$, as claimed. □

6.5. **Extension to multivariate interpolation.** The algorithm presented in Section 6 can be easily extended to a multivariate setting. Therefore, we only give the necessary parameters. We assume $t > \sqrt[m+1]{n(\alpha+2g-1)^m}$ and $g \geqslant 2$. Then we set $\mathcal{R} \stackrel{\text{def}}{=} k/(mn) = (\alpha-g)/(mn)$, $\sigma \stackrel{\text{def}}{=} \lfloor \frac{\ell-g}{\alpha+2g-1} \rfloor$, $b \stackrel{\text{def}}{=} \ell - g + 1 \geqslant a\sigma + 1$. Further we define a set if indices $\{s_i | i = 0, 1, \ldots, m-1\}$ recursively as follows: $s_0 \stackrel{\text{def}}{=} 1$ and $s_i \stackrel{\text{def}}{=} s_{i-1}\sigma + 1$.

In order to ensure $rt > \ell$, we set (assuming $m > 2$)

$$r \stackrel{\text{def}}{=} \left\lceil \frac{\alpha + 3g + m\sqrt[m+1]{n(\alpha+2g-1)^m}}{t - \sqrt[m+1]{n(\alpha+2g-1)^m}} \right\rceil,$$

$$\ell \stackrel{\text{def}}{=} rt - 1.$$

The degree of N gives an easy bound on the list size which is $\leqslant \sigma \cdot \max\{s_1, \cdots, s_{m-1}\}$ $\leqslant (\sigma+1)^m = \Theta(\sigma^m)$. Moreover, the alphabet size can easily be seen to be $Q = q^m$.

6.6. **Consequences.** Since the above construction applies to codes arising from any function field, plugging in the function field with the best possible ratio of g/n, and also using $m \geqslant 2$ correlated functions, we get the following result.

Theorem 6.9 (Main)**.** *For every finite field of size q, with q being a square, an integer $m \geqslant 2$, every $c > 1$, and every $\mathcal{R}$, $\frac{1}{\sqrt{q}-1} < m\mathcal{R} < 1 - \frac{1}{\sqrt{q}-1}$, there is a family of codes over alphabet size q^m of rate $\mathcal{R}$, relative distance at least $1 - m\mathcal{R} - \frac{1}{\sqrt{q}-1}$, and which is list-decodable up to a fraction $1 - c \cdot \left(\frac{3}{\sqrt{q}-1} + m\mathcal{R}\right)^{m/(m+1)}$ of errors using list size at most $O\left(\left(\frac{cm}{c-1}\right)^m \cdot \mathcal{R}^{-m/(m+1)}\right)$. Furthermore, there is a polynomial sized representation of the codes given which encoding and list decoding up to this radius can be performed in polynomial time.*

For decoding up to a fraction $(1-\varepsilon)$ of errors, with the choice $m = \Theta(\log(1/\varepsilon))$ in the above theorem, we get the following.

Corollary 6.10. *For all $\varepsilon > 0$, there is a family of Q-ary codes with $Q = (1/\varepsilon)^{O(\log(1/\varepsilon))}$ which has rate $\Omega(\varepsilon/\log(1/\varepsilon))$ and which is $(1-\varepsilon, (1/\varepsilon)^{O(\log\log(1/\varepsilon))})$-list decodable. Furthermore, the codes have a polynomial sized representation that permits encoding and list decoding up to radius $(1-\varepsilon)$ in polynomial time.*

7. Constructing the representation of codes

We now show how to construct the representation needed for encoding/decoding in polynomial time (as outlined in Section 5.3) for codes based on a tower of function fields proposed by Garcia and Stichtenoth [2]. We begin with the description of this tower of function fields.

Let q_0 be a prime power and $F = \mathbb{F}_{q_0^2}$. The tower of function fields F_i, $i = 0, 1, 2, \ldots$, is defined as a sequence of *Artin-Schreier* extensions. We begin with $F_0 = F(x_0)$, the field of rational functions in x_0. For $i \geqslant 1$, F_i is an algebraic extension of F_{i-1} of degree q_0:

$$F_i = F_{i-1}(x_i) \qquad \text{where } x_i^{q_0} + x_i = \frac{x_{i-1}^{q_0}}{x_{i-1}^{q_0-1} + 1} \ . \tag{17}$$

The above tower meets the Drinfeld-Vlăduţ bound, and thus leads to AG codes with best rate vs. distance trade-offs. In [12], a polynomial time algorithm is presented to compute the generator matrix of such an AG code. All we need to add to this to achieve the representation needed in Section 5.3 are the evaluations of the basis elements at some place R of a specified large degree, and the evaluations of $2g$ extra functions at the code places $P_1, P_2, \ldots, P_n$ and at R. This turns out to be not so straightforward. We begin with a description of some of the basic facts about the function fields F_m. The description assumes some basic knowledge of splitting of places in field extensions.

The genus $g(F_m)$ of F_m satisfies $g(F_m) \leqslant q_0^{m+1}$. Let $\Omega = \{\gamma \in F \mid \gamma^{q_0} + \gamma = 0\}$ denote the set of trace zero elements. For $\theta \in F$, let $P_\theta^{(0)}$ denote the unique zero of $x_0 - \theta$ in F_0. Let $P_\infty^{(0)}$ denote the unique pole of x_0 in F_0. The place $P_\infty^{(0)}$ is totally ramified in the tower, i.e., in each F_m there is precisely one place, $P_\infty^{(m)}$, that lies above $P_\infty^{(0)}$ and moreover this place has degree one. We will use AG codes based on F_m by using as a message space $\text{Ł}((\alpha - 1)P_\infty^{(m)})$.

We now describe the places where the message functions are evaluated for the encoding. Each of the $q_0^2 - q_0$ places $P_\theta^{(0)}$ for $\theta \in F \setminus \Omega$ splits completely in the tower and thus has q_0^m places of degree one lying above it in F_m. Let $n = (q_0^2 - q_0)q_0^m$ and let $P_1, P_2, \ldots, P_n$ be the set of all places of F_m that lie above $P_\theta^{(0)}$ for $\theta \in F \setminus \Omega$. We use the places $P_1, P_2, \ldots, P_n$ as the evaluation places for encoding. Note that $n/g(F_m) \geqslant (q_0 - 1)$ and hence the code meets the Drinfeld-Vlăduţ bound.

Let R_m be the ring of functions that have a pole only at $P_\infty^{(m)}$. As shown in [12], every function R_m has an expression of the form

$$x_0^l \cdot \left(\sum_{e_0=0}^{(m-1)q_0+1} \sum_{e_1=0}^{q_0-1} \cdots \sum_{e_m=0}^{q_0-1} c_{\mathbf{e}} g_0 \frac{x_0^{e_0} x_1^{e_1} \cdots x_m^{e_m}}{\pi_1 \cdots \pi_{m-1}} \right) \tag{18}$$

where $l \geqslant 0$, $c_{\mathbf{e}} \in F$, and for $0 \leqslant k < m$, $g_k = x_k^{q_0-1} + 1$ and $\pi_k = g_0 g_1 \cdots g_k$. Moreover, for any n', Shum et al. [12] present an algorithm running in time polynomial in n', n that outputs a basis of $\text{Ł}(n' P_\infty^{(m)})$ in the above form, together with evaluations of the basis elements at $P_1, P_2, \ldots, P_n$. We note that this latter evaluation part is easily done once the basis elements are represented in the form (18), since for each P_i, evaluating at P_i amounts to substituting appropriate values from $F \setminus \Omega$ for

$x_0, x_1, \ldots, x_m$.[7] Likewise, it suffices to find out the evaluations of $x_0, x_1, \ldots, x_m$ at a place R of degree α in F_m. We now proceed towards this goal, and Theorem 7.2 below asserts that this can be done.

The places of degree α in $F_0 = F(x_0)$ are in one-one correspondence with irreducible polynomials of degree α over F. The place corresponding to an irreducible polynomial $p_0(x_0) \in F[x_0]$ is equal to

$$P_{p_0(x_0)} \stackrel{\text{def}}{=} \left\{ \frac{a(x_0)}{b(x_0)} : a, b \in F[x_0],\ p_0(x_0) | a(x_0),\ p_0(x_0) \nmid b(x_0) \right\}.$$

The following lemma shows that one can find a place of degree α in F_m by finding a place of degree α in F_0 that has a place of degree α lying above it in the extension $[F_m : F_0]$.

Lemma 7.1. *For every $D \geqslant \max\{m+6, 16\}$, there are at least $\frac{q_0^{2D}}{2D \cdot q_0^m}$ places of degree D in F_0 that have a place of degree D lying above them in F_m.*

Proof. Let $g = g(F_m)$ be the genus of F_m; we know $g \leqslant q_0^{m+1}$. Let $\mathcal{T}_D$ denote the set of places in F_m of degree D. By the Hasse-Weil bound, it is known that the number of places $B_D = |\mathcal{T}_D|$ of degree D in F_m satisfies $|B_D - q_0^{2D}/D| < (2+7g)q_0^D/D$; cf. [13, Corollary V.2.10]. It follows that $B_D \geqslant q_0^{2D}/D - 8gq_0^D/D \geqslant q_0^{2D}/D - 8q_0^{m+1+D}/D \geqslant \frac{q_0^{2D}}{2D}$.

Let $\mathcal{T}'_D \subseteq \mathcal{T}_D$ be those places of degree D in F_m that do **not** lie above a place of degree D in F_0. Let $B'_D = |\mathcal{T}'_D|$. The number N_D of places of degree D in F_0 which have a place of degree D lying above them in F_m satisfies $N_D \geqslant (B_D - B'_D)/q_0^m$, since the degree of the extension $[F_m : F_0] = q_0^m$ and so at most q_0^m places of F_m lie above any place of F_0.

Now, if $\tilde{P} \in \mathcal{T}'_D$, then the place $\tilde{P}_0$ lying below it in F_0 must have degree D_0 at most $D/2$ (since D_0 must divide D and is not equal to D). It follows that $B'_D \leqslant q_0^m n_{D/2}$ where $n_{D/2}$ is the number of places of degree at most $D/2$ in F_0. Clearly $n_{D/2} \leqslant \sum_{i=1}^{D/2+1} (q_0^2)^i \leqslant q_0^{D+4}$.

Hence $q_0^m N_D \geqslant B_D - B'_D \geqslant q_0^{2D}/D - 8q_0^{m+1+D}/D - q_0^{m+D+4} \geqslant q_0^{2D}/D - 2q_0^{m+D+4}$ and this latter quantity is easily seen to be at least $\frac{q_0^{2D}}{2D}$ when

$$D \geqslant \max\{m+6, 16\}. \qquad \square$$

We are now ready to prove that the evaluations of the basis functions of $\mathrm{L}(n' P_\infty^{(m)})$ at some place of large degree in F_m can be efficiently found. Recall that the block length n of the code is $n = (q_0^2 - q_0)q_0^m$.

Theorem 7.2. *There is a randomized algorithm that on input integers n', α with $5 \log n \leqslant \alpha \leqslant n$, outputs in expected $\mathrm{poly}(n, n')$ time the evaluations of a set of basis functions of $\mathrm{L}(n' P_\infty^{(m)})$ at some place $R \in \mathbb{P}_{F_m}$ with $\deg(R) = \alpha$.*

Proof. Applying Lemma 7.1, when $5 \log n \leqslant \alpha \leqslant n$, if we pick a monic polynomial $p_0(x_0)$ over $\mathbb{F}_{q_0^2}$ of degree α, then with probability at least $\frac{1}{2\alpha q_0^m} \geqslant \frac{1}{2n^2}$, the degree α place $P_{p_0(x_0)} \in \mathbb{P}_{F_0}$ will have a place of degree α above it in F_m. Suppose that given an irreducible polynomial $p_0(x_0)$ of degree α we could check in $\mathrm{poly}(n)$ time whether the place $P_{p_0(x_0)}$ has some place $R_{p_0(x_0)}$ of degree α above it in F_m, and if

[7] If we begin with $x_0 = \gamma \in F \setminus \Omega$, and solve the equations in (17) in sequence for $x_1, x_2, \ldots, x_m$, then for all solutions, we will have each $x_i \in F \setminus \Omega$; cf. [2, Lemma 3.9].

so, also output the evaluations of $x_0, x_1, \ldots, x_m$ at the place $R_{p_0(x_0)}$. Then we can simply pick a random monic polynomial of degree α, check it is irreducible (which can be done in deterministic polynomial time), and run the above check, and repeat the process until we succeed in finding a place of degree α in F_m together with the evaluations of $x_0, x_1, \ldots, x_m$ at that place. This process will succeed in expected $O(n^2)$ trials of the initial monic polynomial.

Therefore, it remains to check whether a given degree α irreducible $p_0(x_0) \in F[x_0]$ has a place of degree α above it in F_m, and if it does, to find the evaluations of $x_0, \ldots, x_m$ at one of those places. Let $L = F[x_0]/(p_0(x_0))$; L is isomorphic to the finite field $\mathbb{F}_{q_0^{2\alpha}}$. Let $\zeta_0 \in L$ be the residue of $\frac{x_0^{q_0}}{x_0^{q_0-1}+1}$ modulo $p_0(x_0)$. A well-known theorem of Kummer (cf. [13, Theorem III.3.7]), when applied to the tower (17), implies that $P_{p_0(x_0)}$ has some place of degree α above it in F_m iff the sequence of equations $x_1^{q_0} + x_1 = \zeta_0$ and $x_{i+1}^{q_0} + x_{i+1} = \frac{x_i^{q_0}}{x_i^{q_0-1}+1}$ for $1 \leqslant i < m$ has a solution $x_i = \zeta_i \in L$ for $1 \leqslant i \leqslant m$. Moreover, in such a case, there is a place $R_{p_0(x_0)}$ of degree α in F_m such that the evaluation of x_i at the place equals ζ_i for $0 \leqslant i \leqslant m$.

Now, for any $\zeta \in L$ represented in the basis $\{1, x_0, \ldots, x_0^{\alpha-1}\}$ over $\mathbb{F}_{q_0^2}$, one can find all solutions in L of a single equation $z^q + z = \zeta$ in $\mathrm{poly}(\alpha)$ time by solving a linear system with 2α unknowns over $\mathbb{F}_{q_0}$. This is because $z^{q_0} + z$ is a *linearized polynomial* and is a $\mathbb{F}_{q_0}$-linear function on L; cf. [10, Chap. 3, Sec. 4]. It follows that one can find all solutions $(\zeta_1, \ldots, \zeta_m) \in L^m$ to $x_1, \ldots, x_m$ that satisfy the above equations in $q_0^m \cdot \mathrm{poly}(\alpha) = \mathrm{poly}(n)$ time, by solving at most q_0^m linearized polynomial equations. If no such solution exists, the particular choice $p_0(x_0)$ fails. Otherwise, we can use an arbitrary one of those solutions $(\zeta_0, \zeta_1, \ldots, \zeta_m)$ as the evaluations of $x_0, \ldots, x_m$ respectively at a place of degree α. □

8. Extension to list recovering and binary codes

8.1. **List recoverable codes.**

Definition 8.1. A code $C \subseteq \Sigma^n$ is (γ, l, L)-*list recoverable* if for every sequence of sets $S_1, S_2, \ldots, S_n$, where each $S_i \subseteq \Sigma$ has at most l elements, the number of codewords $c \in C$ which satisfy $c_i \in S_i$ for at least γn values of $i \in \{1, 2, \ldots, n\}$ is at most L.

Note a code being (ρ, L)-list decodable is the same thing as it being $(1-\rho, 1, L)$-list recoverable, so the above notion is more general than list decoding. The name list recovering was coined in [3], and this notion has played a crucial role in new constructions of list-decodable codes since.

We now make the following observation. The algorithm in Section 5.2 can be trivially generalized to handle the case when there is a set S_i consisting of possibly more than one triple (y_i, z_{i1}, z_{i2}) for each location i. We simply need to add a constraint for each such triple in the interpolation of Step 2, so that the total number of constraints will now be the total number of triples N (or in other words the total size of all the S_i's). It immediately follows that we get an algorithm for list recovering that works with agreement t as in (5) with N replacing the block length n. Of course, a similar generalization also holds for the m-variate decoding algorithm and the agreement bound of (6). Plugging this into function fields with $g/n = 1/(\sqrt{q}-1)+o(1)$, and performing some straightforward computations, we can get the following results. We note that Corollary 5.11 is a special case obtained by

setting $l = 1$ and $\gamma = \varepsilon$. Corollary 8.3 is obtained using the choice $m = \lceil \log_2(l/\gamma) \rceil$ in Theorem 8.2.

Theorem 8.2. *For all integers $l \geqslant 2$, for all $\gamma > 0$ and all integers $m \geqslant 3$, there is a family of Q-ary codes for $Q = O((ml^{1/m}/\gamma)^{2m^2/(m-1)})$ which has rate $\Omega(\gamma/m^2 \cdot (\gamma/l)^{1/(m-1)})$ and which is (γ, l, L)-list recoverable for*

$$L = O(m^2 \cdot m! \cdot (l/\gamma)^{m/(m-1)}).$$

Moreover, the codes have a natural representation, computable in expected polynomial time, that permits polynomial time encoding as well as polynomial time (γ, l, L)-list recovering.

Corollary 8.3. *For all integers $l \geqslant 2$ and all $\gamma > 0$, there is a family of Q-ary codes for $Q = l^{O(\log\log(l/\gamma))} \cdot (1/\gamma)^{O(\log(1/\gamma))}$ which has rate $\Omega(\gamma/\log^2(l/\gamma))$ and which is (γ, l, L)-list recoverable for $L = (l/\gamma)^{O(\log\log(l/\gamma))}$. Moreover, the codes have a natural representation, computable in expected polynomial time, that permits polynomial time encoding as well as polynomial time (γ, l, L)-list recovering.*

A similar generalization of the algorithm in Section 6.2 is possible.

8.2. **Binary codes for list decoding up to radius $(1/2 - \varepsilon)$.** We now consider the problem of constructing binary codes for list decoding up to radius $(1/2 - \varepsilon)$, for small $\varepsilon > 0$. Using the list recoverable codes of Corollary 8.3 with parameters $l = O(1/\varepsilon^2)$ and $\gamma = \varepsilon/2$ as the outer code in a concatenation scheme with a constant-sized binary inner code with Q codewords and rate $\Omega(\varepsilon^2)$ and that is $(1/2 - \varepsilon/2, l)$-list decodable, we can show the following.

Theorem 8.4. *For every $\varepsilon > 0$, there is a family of binary codes of rate $\Omega(\varepsilon^3/\log^2(1/\varepsilon))$ that is $(1/2 - \varepsilon, (1/\varepsilon)^{O(\log\log(1/\varepsilon))})$-list-decodable. The codes can be constructed in expected polynomial time and admit a polynomial time encoding algorithm as well a polynomial time list decoding algorithm for radius $(1/2 - \varepsilon)$.*

Generalizing the algorithm from Section 6.2, we can achieve a better rate of $\Omega(\varepsilon^3/\log(1/\varepsilon))$ though the construction needs some preprocessed information.

We remark that the recent construction of [4] achieves a rate of $\Omega(\varepsilon^3)$ for $(1/2 - \varepsilon, L)$-list-decodable codes, but their construction time as well as list size L is $n^{\Omega(1/\varepsilon^3)}$. In contrast, our codes are uniformly constructive, i.e., can be constructed and decoded in time $f(\varepsilon)n^{O(1)}$ with exponent of n independent of ε, and achieve a list size independent of the block length.

9. Concluding remarks

We have generalized the Parvaresh-Vardy approach to all algebraic-geometric codes. These new codes are obtained by evaluating several functions from a function field, which are correlated in a carefully specified way, at some rational points on the algebraic curve. Some complications arise in the higher genus case compared to RS codes (the genus 0 case), but we showed how to handle these with a minor loss in error-correction performance.

The scheme of evaluating correlated functions/messages to perform the encoding is quite general and can also be applied to Chinese Remainder codes (in fact for these codes there is a precise parallel with Reed-Solomon codes), and more generally to "ideal-based" codes [5]. Details are quite straightforward now that we have abstracted the salient features of the algorithm for general AG-codes.

Acknowledgments

We thank Farzad Parvaresh and Alexander Vardy for sending us an early draft of their paper. The first author thanks Henning Stichtenoth for useful discussions on the splitting behavior of places in towers of function fields. The second author thanks Felipe Voloch for helpful comments and suggestions. We also thank an anonymous referee for his/her valuable comments.

References

1. Arnaldo Garcia and Henning Stichtenoth, *A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlădut bound*, Inventiones Mathematicae **121** (1995), 211–222. MR1345289 (96d:11074)
2. ———, *On the asymptotic behavior of some towers of function fields over finite fields*, Journal of Number Theory **61** (1996), no. 2, 248–273. MR1423052 (97i:11067)
3. Venkatesan Guruswami and Piotr Indyk, *Expander-based constructions of efficiently decodable codes*, Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS), 2001, pp. 658–667. MR1948755
4. Venkatesan Guruswami and Atri Rudra, *Explicit capacity-achieving list-decodable codes*, Proceedings of the 38th ACM Symposium on Theory of Computing, May 2006, pp. 1–10. MR2277125
5. Venkatesan Guruswami, Amit Sahai, and Madhu Sudan, *Soft-decision decoding of Chinese Remainder codes*, Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS), 2000, pp. 159–168. MR1931814
6. Venkatesan Guruswami and Madhu Sudan, *Improved decoding of Reed-Solomon and algebraic-geometric codes*, IEEE Transactions on Information Theory **45** (1999), 1757–1767. MR1720630 (2000j:94033)
7. Venkatesan Guruswami and Madhu Sudan, *On representations of algebraic-geometric codes*, IEEE Transactions on Information Theory **47** (May 2001), no. 4, 1610–1613. MR1830110 (2002b:94046)
8. Ralf Koetter and Alexander Vardy, *Soft decoding of Reed Solomon codes and optimal weight assignments*, ITG Fachtagung (Berlin, Germany), January 2002.
9. ———, *Algebraic soft-decision decoding of Reed-Solomon codes*, IEEE Transactions on Information Theory **49** (2003), no. 11, 2809–2825. MR2027561 (2004k:94093)
10. Rudolf Lidl and Harald Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press, Cambridge, MA, 1986. MR860948 (88c:11073)
11. Farzad Parvaresh and Alexander Vardy, *Correcting errors beyond the Guruswami-Sudan radius in polynomial time*, Proceedings of the 46th IEEE Symposium on Foundations of Computer Science (FOCS), 2005, pp. 285–294.
12. Kenneth Shum, Ilia Aleshnikov, P. Vijay Kumar, Henning Stichtenoth, and Vinay Deolalikar, *A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound*, IEEE Transactions on Information Theory **47** (2001), no. 6, 2225–2241. MR1873198 (2003e:94110)
13. Henning Stichtenoth, *Algebraic function fields and codes*, Universitext, Springer-Verlag, Berlin, 1993. MR1251961 (94k:14016)
14. Madhu Sudan, *Decoding of Reed-Solomon codes beyond the error-correction bound*, Journal of Complexity **13** (1997), no. 1, 180–193. MR1449766 (98f:94024)
15. Michael A. Tsfasman, Serge G. Vlădut, and Thomas Zink, *Modular curves, Shimura curves, and codes better than the Varshamov-Gilbert bound*, Math. Nachrichten **109** (1982), 21–28. MR705893 (85i:11108)

Department of Computer Science & Engineering, University of Washington, Seattle, Washington 98195

E-mail address: venkat@cs.washington.edu

Department of Computer Science, University of Texas at Austin, Austin, Texas 78712

E-mail address: anindya@cs.utexas.edu