Folded Codes from Function Field Towers and Improved Optimal Rate List Decoding
(Extended Abstract)

Venkatesan Guruswami*
Computer Science Department.
Carnegie Mellon University
Pittsburgh, PA, USA
guruswami@cmu.edu

Chaoping Xing †
Division of Mathematical Sciences
School of Physical & Mathematical Sciences
Nanyang Technological University, Singapore
xingcp@ntu.edu.sg

ABSTRACT
We give a new construction of algebraic codes which are efficiently list decodable from a fraction $1 - R - \varepsilon$ of adversarial errors where $R$ is the rate of the code, for any desired positive constant $\varepsilon$. The worst-case list size output by the algorithm is $O(1/\varepsilon)$, matching the existential bound for random codes up to constant factors. Further, the alphabet size of the codes is a constant depending only on $\varepsilon$ — it can be made $\exp(O(1/\varepsilon^2))$ which is not much worse than the non-constructive $\exp(1/\varepsilon)$ bound of random codes. The code construction is Monte Carlo and has the claimed list decoding property with high probability. Once the code is (efficiently) sampled, the encoding/decoding algorithms are deterministic with a running time $O(N)$ for an absolute constant $c$, where $N$ is the code’s block length.

Our construction is based on a careful combination of a linear-algebraic approach to list decoding folded codes from towers of function fields, with a special form of subspace-evasive sets. Instantiating this with the explicit “asymptotically good” Garcia-Stichtenoth (GS for short) tower of function fields yields the above parameters. To illustrate the method in a simpler setting, we also present a construction based on Hermitian function fields, which offers similar guarantees with a list-size and alphabet size polylogarithmic in the block length $N$.

*Research supported in part by a Packard Fellowship and NSF CCF 0963975. Some of this work was done during a visit to Nanyang Technological University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

†Research supported by the Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03 and Singapore A*STAR SERC under Research Grant 1121720011.

In comparison, algebraic codes achieving the optimal tradeoff between list decodability and rate based on folded Reed-Solomon codes have a decoding complexity of $N^{O(1/\varepsilon^2)}$, an alphabet size of $N^{O(1/\varepsilon^2)}$, and a list size of $O(1/\varepsilon^2)$ (even after combination with subspace-evasive sets). Thus we get an improvement over the previous best bounds in all three aspects simultaneously, and are quite close to the existential random coding bounds. Along the way, we shed light on how to use automorphisms of certain function fields to enable list decoding of the folded version of the associated algebraic-geometric codes.

Categories and Subject Descriptors
E.4 [Data]: Coding and Information Theory

General Terms
Theory

Keywords
List error-correction, Algebraic codes, Subspace-evasive sets, Pseudorandomness, List decoding capacity.

1. INTRODUCTION
An error-correcting code $C$ of block length $N$ over an alphabet $\Sigma$ maps a set $M$ of messages into codewords in $\Sigma^N$. The rate of the code $C$, denoted $R$, equals $\frac{1}{N} \log_{|\Sigma|} |M|$. In this work, we will be interested in codes for adversarial noise, where the channel can arbitrarily corrupt up to a fraction $\tau$ of the $N$ symbols. The goal will be to correct such errors and recover the original message/codeword. It is easy to see that information-theoretically, we need to receive at least $RN$ symbols correctly in order to recover the message (since $|M| = |\Sigma|^{RN}$), so we must have $\tau \leq 1 - R$.

Perhaps surprisingly, in a model called list decoding, recovery up to this information-theoretic limit becomes possible. Let us say that a code $C \subseteq \Sigma^N$ is $(\ell, \tau)$-list decodable if for every received word $y \in \Sigma^N$, there are at most $\ell$ codewords $c \in C$ such that $y$ and $c$ differ in at most $\tau N$ positions. Such a code allows, in principle, the correction of a fraction $\tau$ of errors, outputting at most $\ell$ candidate codewords one of which is the originally transmitted codeword.

The probabilistic method shows that a random code of rate $R$ over an alphabet of size $\exp(O(1/\varepsilon))$ is with high
probability $(1 - R - \varepsilon, O(1/\varepsilon))$-list decodable [2]. However, it is not known how to construct or even randomly sample such a code for which the associated algorithmic task of list decoding (i.e., given $\gamma \in \Sigma^N$, find the list of codewords within fractional radius $1 - R - \varepsilon$) can be performed efficiently. This work takes a big step in that direction, giving a randomized construction of such efficiently list-decodable codes over a slightly worse alphabet size of $\exp(O(1/\varepsilon^2))$. We note that the alphabet size needs to be at least $\exp(\Omega(1/\varepsilon))$ in order to list decode from a fraction $1 - R - \varepsilon$ of errors, so this is close to optimal. For the list-size needed as a function of $\varepsilon$ for decoding a $1 - R - \varepsilon$ fraction of errors, the best lower bound is only $\Omega(\log(1/\varepsilon))$ [7], but as mentioned above, even random coding arguments only achieve a list-size of $O(1/\varepsilon)$, which our construction matches up to constant factors.

We now review some of the key results on algebraic list decoding leading up to this work. A more technical comparison with related work appears in Section 1.1. The first construction of codes that achieved the optimal trade-off between rate and list-decoding radius, i.e., enabled list decoding up to a fraction $1 - R - \varepsilon$ of worst-case errors with rate $R$, was due to Guruswami and Rudra [8]. They showed that a variant of Reed-Solomon (RS) codes called folded RS codes admit such a list decoder. For a decoding radius of $1 - R - \varepsilon$, the code was based on bundling together disjoint windows of $m = \Theta(1/\varepsilon^2)$ consecutive symbols of the RS codeword into a single symbol over a larger alphabet. As a result, the alphabet size of the construction was $N^{O(1/\varepsilon^2)}$. Ideas based on code concatenation and expander codes can be used to bring down the alphabet size to $\exp(O(1/\varepsilon^4))$, but this compromises some nice features such as list recovery and soft decoding of the folded RS code. Also, the decoding time complexity as well as proven bound on worst-case output list size for these constructions were $N^{O(1/\varepsilon)}$ which is rather large.

Our main final result statement is the following. The codes we construct are a randomly sampled subcode of an analog of folded RS codes for an asymptotically optimal tower of function fields due to Garcia and Stichtenoth [3, 4].

**Theorem 1.1.** For any $R \in (0,1)$ and positive constant $\varepsilon \in (0,1)$, there is a Monte Carlo construction of a family of codes of rate at least $R$ over an alphabet size $\exp(O((1/\varepsilon^2)))$ that are encodable and $(1 - R - \varepsilon, O(1/\varepsilon))$-list decodable in $\poly(N)$ time, where $N$ is the block length of the code.

To illustrate our ideas in an algebraically simpler setting, we first give a construction based on a tower of Hermitian field extensions [12]. This gives a similar result, albeit with alphabet size and list-size upper bound polylogarithmic in $N$.

### 1.1 Prior and related work

Let us recap a bit more formally the construction of folded RS codes from [8]. One begins with the RS encoding of a polynomial $f \in \mathbb{F}_q[X]$ of degree $< k$ with the evaluation points ordered as $1, \gamma, \ldots, \gamma^{n-1}$ for some primitive element $\gamma \in \mathbb{F}_q$ and $n < q$. For an integer “folding” parameter $m \geq 1$ that divides $n$, the folded RS codeword is defined over alphabet $\mathbb{F}_q^m$ and consists of $n/m$ blocks, with the $j$th block consisting of the $m$-tuple $(f(\gamma^{j-1})), f(\gamma^{(j-1)+1}), \ldots, f(\gamma^{j-1}))$. The algorithm in [8] for list decoding these codes was based on the algebraic identity $f(\gamma X) = \mathcal{F}(X)^m$ in the residue field $\mathbb{F}_q[X]/(X^{q^m-1} - \gamma)$ where $\mathcal{F}$ denotes the residue $f$ mod $(X^{q^m-1} - \gamma)$. This identity is used to solve for $f$ from an equation of the form $Q(X, f(X), f(\gamma X), \ldots, f(\gamma^{m-1} X)) = 0$ for some low-degree nonzero multivariate polynomial $Q$. The high degree $q > n$ of this identity, coupled with $s \approx 1/\varepsilon$, led to the large bounds on list-size and decoding complexity in [8].

One possible approach to reduce $q$ (as a function of the code length) in this construction would be to work with algebraic-geometric codes based on function fields $K$ over $\mathbb{F}_q$ with more rational points. However, an automorphism $\sigma$ of $K$ that can play the role of the automorphism $f(X) \mapsto f(\gamma X)$ of $\mathbb{F}_q(X)$ is only known (or even possible) for very special function fields. This approach was used in [5] to construct list-decodable codes based on cyclotomic function fields using as $\sigma$ certain Frobenius automorphisms. These codes improved the alphabet size to polylogarithmic in $N$, but the bound on list-size and decoding complexity remained $N^{O(1/\varepsilon)}$.

Recently, a linear-algebraic approach to list decoding folded RS codes has been discovered in [16, 6]. Here, in the interpolation stage, which is common to all list decoding algorithms for algebraic-geometric codes [15, 9, 11, 8], following the idea in [16] one finds a linear multivariate polynomial $Q(X,Y_1,\ldots,Y_s)$ whose total degree in the $Y_i$’s is 1. The simple but key observation driving [6] is that the equation $Q(X, f(X), \ldots, f(\gamma^{m-1} X)) = 0$ now becomes a linear system in the coefficients of $f$. Further, it is shown that the solution space has dimension less than $s$, which again gives a list-size upper bound of $q^{s-1}$. Finally, since the list of candidate messages fall in an affine space, it was noted in [6] that one can bring down the list size by carefully “pre-coding” the message polynomials so that their $k$ coefficients belong to a “subspace-evasive set” (which has small intersection with every $s$-dimensional subspace of $\mathbb{F}_q^s$). This idea was used in [6] to give a randomized construction of $(1 - R - \varepsilon, O(1/\varepsilon^2))$-list decodable codes of rate $R$ (in fact, the list size bound is worse — it is $\Omega(N)$ if one requires efficient encoding of the code). However, the alphabet size and runtime of the decoding algorithm both remained $N^{O(1/\varepsilon)}$. In [10], similar results were also shown for derivative codes, where the encoding of a polynomial $f$ consists of the evaluations of $f$ and its first $m - 1$ derivatives at distinct field elements.

In a concurrent independent work, Dvir and Lovett gave an elegant construction of explicit subspace-evasive sets based on certain algebraic varieties [1]. This yields an explicit version of the codes from [6], albeit with a worse list size bound of $(1/\varepsilon)^{O(1/\varepsilon)}$. This work and [1] are incomparable in terms of results. The big advantage of [1] is the deterministic construction of the code. The benefits in our work are (i) our list-size of $O(1/\varepsilon)$ is much better and in fact optimal up to constant factors, and (ii) we are able to construct codes over an alphabet size that is a constant independent of $N$, whereas in [1] the $N^{O(1/\varepsilon^2)}$ alphabet size of folded RS codes is inherited. Both our work and [1] achieve a decoding complexity of $O_{\varepsilon}(N^s)$ with exponential independent of $\varepsilon$.

We should note that since we require sets that are evasive with respect to subspaces of large dimension, and which have
further structural properties needed in the decoding, we cannot use the construction in [1] to make the codes in this work explicit.

### 1.2 Our techniques

We describe some of the main new ingredients that go into our work. We need both new algebraic insights and constructions, as well as ideas in pseudorandomness relating to subspace-evasive sets with additional structure. We describe these in turn below.

**Algebraic ideas.** As mentioned above, effecting the original “non-linear” approach in [8, 5] with automorphisms of more general function fields seems intricate at best. The correct generalization of the linear-algebraic list decoding approach to the function field case is also not obvious. One of the main algebraic insights in this work is noting that the right way to generalize the linear-algebraic approach to codes based on algebraic function fields is to rely on the local power series expansion of functions from the message space at a suitable rational point. (The case for Reed-Solomon codes being the expansion around 0, which is a finite polynomial form.)

Working with a suitable automorphism which has a “diagonal” action on the local expansion lets us extend the linear-algebraic decoding method to AG codes. Implementing this for specific AG codes requires an explicit specification of a basis for an associated message (Riemann-Roch) space, and the efficient computation of the local expansion of the basis elements at a special rational point on the curve. We show how to do this for two towers of function fields: the Hermitian tower [12] and the asymptotically optimal Garcia-Stichtenoth tower [3, 4]. The former tower is quite simple to handle: it has an easily written down explicit basis, and we show how to compute the local expansion of functions around the point with all zero coordinates. However, the Hermitian tower does not have bounded ratio of the genus to number of rational points, and does not give constant alphabet codes (we can get codes over an alphabet size that is polylogarithmic in the block length though). Explicit basis for Riemann-Roch spaces of the Garcia-Stichtenoth tower were constructed in [13]. Regarding local expansions, one major difference is that we work with local expansion of functions at the point at infinity, which is fully “ramified” in the tower. For both these towers, we find and work with a nice automorphism that acts diagonally on the local expansion, and use it for folding the codes and decoding them by solving a linear system.

**Pseudorandomness.** These algebraic ideas enable us to pin down the messages into a subspace of dimension linear in the message length. To prune this list, we need several additional ideas. The starting point is to follow [6] and only encode messages in a subspace-evasive set which has small intersection with low-dimensional subspaces. Implementing this in our case, however, leads to several problems. First, since the subspace we like to avoid intersecting much has large dimension, the list size bound will be linear in the code length and not a constant like in our final result. More severely, we cannot go over the elements of this subspace to prune the list as that would take exponential time. To solve the latter problem, we observe that the subspace has a special “periodic” structure, and exploit this to show the existence of large “hierarchically subspace evasive” (h.s.e) subsets which have small intersection with the projection of the subspace on certain prefixes. Isolating the periodic property of the subspaces, and formulating the right notion of evasiveness w.r.t to such subspaces, is an important aspect of this work.

We also give a pseudorandom construction of good h.s.e sets using limited wise independent sample spaces, in a manner enabling the efficient iterative computation of the final list of intersecting elements. With some additional ideas, we ensure that one can efficiently index into a large subset of our h.s.e set construction (this is needed to get an efficient encoding algorithm for our code). As a further ingredient, we note that the number of possible subspaces that arise in the decoding is much smaller than the total number of possibilities. Using this together with a trick to take the intersection of two subspace evasive set constructions, we are able to reduce the list size to a constant.

### 1.3 Organization

We begin by isolating the special notion of subspaces which our evasive sets should avoid intersecting too much (Section 2). We describe our construction of folded Hermitian codes and a linear-algebraic list decoding algorithm for these codes in Section 3. In Section 4, we define and construct the special “hierarchically” subspace-evasive (h.s.e) sets that we need (the proofs are omitted and appear in the full version). We show how to combine the h.s.e sets with folded Hermitian codes in Section 5; this gives a result similar to Theorem 1.1 with polylogarithmic alphabet and list size. We show how similar ideas can be used to construct folded codes based on the Garcia-Stichtenoth tower, and how to combine them with h.s.e sets to get our main result (Theorem 1.1) in Section 6.

## 2. PERIODIC SUBSPACES

The list decoding algorithm for our algebraic codes will first pin down the candidate messages to a subspace. The structure of the subspace will be important to us in order to be able to efficiently prune it to a much smaller list. In this section, we make some important definitions capturing this property. Let us begin with some notation.

**Notation 1. (Projection of vectors and sets)** For a vector \( y = (y_1, y_2, \ldots, y_m) \in \mathbb{F}_q^m \) and positive integers \( t_1 \leq t_2 \leq m \), we denote by \( \text{proj}_{[t_1, t_2]}(y) \in \mathbb{F}_q^{2-t_1+1} \) its projection onto coordinates \( t_1 \) through \( t_2 \), i.e., \( \text{proj}_{[t_1, t_2]}(y) = (y_{t_1}, y_{t_1+1}, \ldots, y_{t_2}) \). When \( t_1 = 1 \), we use \( \text{proj}_1(y) \) to denote \( \text{proj}_{[1, t_2]}(y) \).

For a subset \( S \subseteq \mathbb{F}_k^d \) and positive integers \( t_1 \leq t_2 \leq k \), we denote the projection of \( S \) onto the coordinates in the range \([t_1, t_2]\) by \( \text{proj}_{[t_1, t_2]}(S) \). Formally, \( \text{proj}_{[t_1, t_2]}(S) = \{ \text{proj}_{[t_1, t_2]}(y) \mid y \in S \} \). Again, we use \( \text{proj}_1(S) \) to denote \( \text{proj}_{[1, t_2]}(S) \).

The specific definition of periodic subspaces below (which might appear rather technical) is motivated by the structure of the subspaces arising in our list decoding application (for example, as guaranteed by Lemma 3.7). The special structure of these subspaces is important to guarantee the existence of “subspace evasive sets” (defined later) that are good enough for our purposes.

**Definition 1. (s, \( \Delta \))-periodic subspaces** For positive integers \( s, \Delta, b \), an affine subspace \( W \) of \( \mathbb{F}_2^k \) where \( k = 2^b \) according to the definition, an \( (s, \Delta) \)-periodic subspace is...
bΔ is said to be \((s, \Delta)\)-periodic if there is a subspace \(U\) of \(F_q^n\) of dimension less than \(s\) such that for all \(y \in W \) and \(1 \leq i \leq b\), \(\text{proj}_{(i-1)\Delta+1,i\Delta}(y)\) belongs to the affine space \(U + b_i\), where \(b_i\) is a column vector whose coordinates are affine combinations (depending only on \(i\)) of the first \((i-1)\Delta\) coordinates of \(y\); formally, \(b_i = C_i \cdot \text{proj}_{i\Delta}(y) + v_i\), for some matrix \(C_i \in F_q^{(i-1)\Delta}\) and \(v_i \in F_q^\Delta\). We can represent such an affine subspace by \(U \cup \{C_i, v_i\}_{i=1}^b\).

Note that if \(W\) is an \((s, \Delta)\)-periodic subspace of \(F_q^n\), for every \(i, 1 \leq i \leq b\), and every \(a \in F_q^{(i-1)\Delta}\), the affine space \(\{\text{proj}_{(i-1)\Delta+1,i\Delta}(w) \mid w \in W \land \text{proj}_{i\Delta}(w) = a\}\) has dimension at most \(s\) (and in particular has at most \(q^s\) elements). Therefore, by an inductive argument, we have \(\text{proj}_{i\Delta}(W) \leq q^s\) for \(1 \leq i \leq b\), which together with the fact that \(\text{proj}_{i\Delta}(W)\) is an affine subspace implies the following.

**Observation 2.1.** If \(W\) is an \((s, \Delta)\)-periodic subspace of \(F_q^n\), then for \(i = 1, 2, \ldots, b\), \(\text{proj}_{i\Delta}(W)\) is also \((s, \Delta)\)-periodic and has dimension at most \(s-1\) as an affine subspace of \(F_q^n\).

### 3. Folded Codes from the Hermitean Tower

In this section, we will describe a family of folded codes based on the Hermitean function field (or rather a tower of such fields).

#### 3.1 Background on Hermitean tower

In what follows, let \(r\) be a prime power and let \(q = r^2\). We denote by \(F_r\) the finite field with \(q\) elements. The Hermitean function tower that we are going to use for our code construction was discussed in [12]. The reader may refer to [12] for the detailed background on the Hermitean function tower, and Stichtenoth's book [14] for general background on algebraic function fields and their use in constructing algebraic-geometric codes. The Hermitean tower is defined by the following recursive equations

\[
x^{r^i}_{i+1} + x^{r^i}_{i+1} = x^{r^i+1}_{i+1}, \quad i = 1, 2, \ldots, e - 1.
\]

Put \(F_r = F_q(x_1, x_2, \ldots, x_e)\) for \(e \geq 2\). We will assume that \(r \geq 2e\).

**Rational places.** The function field \(F_r\) has \(r^{e+1} + 1\) rational places. One of these is the “point at infinity” which is the unique pole \(P_{\infty}\) of \(x_1\) (and is fully ramified). The other \(r^{e+1}\) come from the rational places lying over the unique zero \(P_0\) of \(x_1 - \alpha\) for each \(\alpha \in F_q\). Note that for every \(\alpha \in F_q, P_0\) splits completely in \(F_r\), i.e., there are \(r^{e+1}\) rational places lying over \(P_0\). Intuitively, one can think of the rational places of \(F_r\) (besides \(P_{\infty}\)) as being given by \(e\)-tuples \((\alpha_1, \alpha_2, \ldots, \alpha_e) \in F_q^e\) that satisfy \(\alpha_{i+1}^{r^i} + \alpha_{i+1} = \alpha_{i+1}^{r^i+1}\) for \(i = 1, 2, \ldots, e - 1\). For each value of \(\alpha \in F_q\), there are precisely \(r\) solutions to \(\beta^{r^i} + \beta = \alpha^{r^i+1}\), so the number of such \(e\)-tuples is \(r^{e+1}\) (\(r^2\) choices for \(\alpha_1\), and then \(r\) choices for each successive \(\alpha_i, 2 \leq i \leq e\)).

**Riemann-Roch spaces.** For a place \(P\) of \(F_r\), we denote by \(\nu_P\) the discrete valuation of \(P\); for a function \(h \in F_r\), if \(h\) has a zero at \(P\), then \(\nu_P(h)\) gives the number (multiplicity) of zeroes, if \(h\) has a pole at \(P\), then \(\nu_P(h)\) gives the pole in fact an affine space. For convenience, we blur this distinction, which is not too important for us, and use the terminology periodic subspace to refer to them.

Then the dimension \(\ell(l(P_{\infty}))\) is at least \(l - q_{e+1} + 1\) and furthermore, \(\ell(l(P_{\infty})) = l - q_{e+1} + 1\) if \(l \geq 2q_{e+1} - 1\). A basis over \(F_q\) of \(\ell(l(P_{\infty}))\) can be explicitly constructed as follows

\[
x^{r^i_1} \cdots x^{r^i_e} : (j_1, \ldots, j_e) \in \mathbb{Z}_{\geq 0}, \sum_{i=1}^e j_i (r^{e-1}(r+1)^{-1}) \leq l
\]

We stress that evaluating elements of \(\ell(l(P_{\infty}))\) at the rational places of \(F_r\) (other then \(P_{\infty}\)) is easy: we simply have to evaluate a linear combination of the monomials allowed in (1) at the tuples \((\alpha_1, \alpha_2, \ldots, \alpha_e) \in F_q^e\) mentioned above. In other words, it is just evaluating an \(e\)-variate polynomial at a specific subset of \(r^{e+1}\) points of \(F_q^e\), and can be accomplished in polynomial time.

**Genus.** The genus \(g_e\) of the function field \(F_r\) is given by

\[
g_e = \frac{1}{2} \left( \sum_{i=1}^e r^i (1 + \frac{1}{r})^{-1} - (r+1)^{-1}\right)\leq \frac{1}{2} \sum_{i=1}^e \frac{1}{r^{i+1}} \leq \frac{1}{2} \sum_{i=1}^e \frac{1}{r^{i+1}} \leq e r^e
\]

where the last step used \(r \geq 2e\).

**A useful automorphism.** Let \(\gamma\) be a primitive element of \(F_q\), and consider the automorphism \(\sigma \in \text{Aut}(F_r/F_q)\) defined by

\[
\sigma : x_i \mapsto x_i^{r^{e+1}-1}x_i \quad \text{for } i = 1, 2, \ldots, e.
\]

The order of \(\sigma\) is \(q - 1\) and furthermore, we have the following facts:

(i) Let \(P_0\) be the unique common zero of \(x_1, x_2, \ldots, x_e\) (this corresponds to the \(e\)-tuple \((0, 0, \ldots, 0)\)), and \(P_{\infty}\) the unique pole of \(x_1\). The automorphism \(\sigma\) keeps \(P_0\) and \(P_{\infty}\) unchanged, i.e., \(P_0^\sigma = P_0\) and \(P_{\infty}^\sigma = P_{\infty}\).

(ii) Let \(\Pi\) be the set of all the rational places which are neither \(P_{\infty}\) nor zeros of \(x_1\). Then \(|\Pi| = (q - 1)^{r^e}\).

Moreover, \(\sigma\) divides \(\Pi\) into \(r^{e+1}\) orbits and each orbit has \(q - 1\) places. For an integer \(m\) with \(1 \leq m < q^{e+1}\), we can label \(N\) distinct elements \(P_1, P_1^{r^e}, \ldots, P_1^{r^{e+1}-1}\), \(\ldots, P_N, P_N^{r^e}, \ldots, P_N^{r^{e+1}-1}\) in \(\Pi\), as long as \(N \leq r^{e+1} \left\lfloor \frac{q_{e+1}}{m} \right\rfloor\).

**Definition 2.** (Folded codes from the Hermitean tower) Assume that \(m, l, N\) are positive integers satisfying \(1 \leq m < q - 1\) and \(l/m \leq N \leq r^{e+1} \left\lfloor \frac{q_{e+1}}{m} \right\rfloor\). The folded code from \(F_r\) with parameters \(N, l, q, e, m\), denoted by \(\text{FH}(N, l, q, e, m)\), encodes a message function \(f \in \ell(l(P_{\infty}))\) as

\[
f \mapsto \left[ \begin{array}{c} f(P_1) \\ f(P_1^r) \\ \vdots \\ f(P_1^{r^{e+1}-1}) \\ f(P_2) \\ f(P_2^r) \\ \vdots \\ f(P_2^{r^{e+1}-1}) \\ \vdots \\ f(P_N) \\ f(P_N^r) \\ \vdots \\ f(P_N^{r^{e+1}-1}) \end{array} \right] \in (F_q^n)^N.
\]

**Lemma 3.1.** The above code \(\text{FH}(N, l, q, e, m)\) is an \(F_q\)-linear code over alphabet size \(q^m\), rate at least \(\frac{l}{r^{e+1} - 1} \geq \frac{1}{N}\), and minimum distance at least \(N - \frac{1}{m}\).
3.2 Redefining the code in terms of local expansion at \( P_0 \)

For our decoding, we will actually recover the message \( f \in L(I(P_{\infty})) \) in terms of the coefficients of its power series expansion around \( P_0 \)

\[
f = f_0 + f_1 x + f_2 x^2 + \cdots
\]

where \( x := x_1 \) is the local parameter at \( P_0 \) (which means that \( x_1 \) has exactly one zero at \( P_0 \), i.e., \( \nu_{P_0}(x_1) = 1 \)). In fact, realizing that one must work in this power series representation is one of the key insights in this work.

Let us first show that one can efficiently move back-and-forth between the representation of \( f \in L(I(P_{\infty})) \) in terms of a basis for \( L(I(P_{\infty})) \) and its power series representation \((f_0, f_1, \ldots)\) around \( P_0 \). Since the mapping \( f \mapsto (f_0, f_1, \ldots) \) is \( F_q \)-linear, it suffices to compute the local expansion at \( P_0 \) of a basis for \( L(I(P_{\infty})) \). By the structure of the basis functions in (1), it is sufficient to find an algorithm of efficiently finding local expansions of \( x_i \) at \( P_0 \) for every \( i = 1, 2, \ldots, e \). This can be done inductively leading to the following lemma (proof skipped and deferred to the full version).

**Lemma 3.2.** For any \( n \), one can compute the first \( n \) terms of the local expansion of the basis elements (1) at \( P_0 \) using \( \text{poly}(n) \) operations over \( F_q \).

To keep the list output by the algorithm at a controllable size, we will combine the code with certain special subspace evasive sets. For this purpose, we will actually need to index the messages of the code by the first \( k \) coefficients \((f_0, f_1, \ldots, f_{s-1})\) of the local expansion of the function \( f \) at \( P_0 \). This requires that for every \((f_0, f_1, \ldots, f_{s-1})\) there is a \( f \in L(I(P_{\infty})) \) whose power series expansion has the \( f_i \) as the first \( k \) coefficients. This is easy to ensure by taking \( l = k + 2g_e - 1 \) as we argue below. Note that to ensure that \( L(I(P_{\infty})) \) has dimension \( k \), it suffices to pick \( l = k + g_e - 1 \) by the Riemann-Roch theorem. We pick \( l \) to be \( g_e \) more than this bound. Since the genus will be much smaller than the code length, we can afford this small loss in parameters.

Let us define the local expansion map \( \text{ev}_{P_0} : L((k+2g_e-1)P_0) \to F_q^k \) that maps \( f \) to \((f_0, f_1, \ldots, f_{s-1})\) where \( f = f_0 + f_1 x + f_2 x^2 + \cdots \) is the local expansion of \( f \) at \( P_0 \).

**Claim 3.3.** \( \text{ev}_{P_0} \) is an \( F_q \)-linear surjective map. Further, we can compute \( \text{ev}_{P_0} \) using \( \text{poly}(k, g_e) \) operations over \( F_q \) given a representation of the input \( f \in L((k+2g_e-1)P_{\infty}) \) in terms of the basis (1).

**Proof.** The \( F_q \)-linearity of \( \text{ev}_{P_0} \) is clear. The kernel of \( \text{ev}_{P_0} \) is \( L((k+2g_e-1)P_0 - kP_0) \) which has dimension exactly \( g_e \) by the Riemann-Roch theorem. By the rank-nullity theorem, the image must have dimension \( k \), and so the map is surjective. The claimed complexity of computation follows immediately from Lemma 3.2.

For each \((f_0, f_1, \ldots, f_{s-1}) \in F_q^k \), we can therefore pick a pre-image in \( L((k+2g_e-1)P_0) \). For convenience, we will denote an injective map making such a unique choice by \( \kappa_{P_0} : F_q^k \to L((k+2g_e-1)P_0) \). By picking the pre-images of a basis of \( F_q^k \) and extending it by linearity, we can assume \( \kappa_{P_0} \) to be \( F_q \)-linear, and thus specify it by a \((k+g_e) \times k \) matrix. We record this fact for easy reference below.

**Claim 3.4.** The map \( \kappa_{P_0} : F_q^k \to L((k+2g_e-1)P_{\infty}) \) is \( F_q \)-linear and injective. We can compute a representation of this linear transformation using \( \text{poly}(k, g_e) \) operations over \( F_q \), and the map itself can be evaluated using \( \text{poly}(k, g_e) \) operations over \( F_q \).

We will now redefine a version of the folded Hermitian code that maps \( F_q^k \) to \((F_q^m)^N\) by composing the folded encoding (3) from the original Definition 2 with \( \kappa_{P_0} \).

**Definition 3.** (Folded Hermitian code using local expansion) The folded Hermitian code \( \text{FH}(N, k, q, e, m) \) maps \( f = (f_0, f_1, \ldots, f_{N-1}) \in F_q^k \) to \( \text{FH}(N, k+2g_e-1, q, e, m)(\kappa_{P_0}(f)) \in (F_q^m)^N \).

The rate of the above code equals \( k/(Nm) \) and its distance is at least \( N - (k+2g_e-1)/m \).

3.3 List decoding folded codes from the Hermitian tower

We now present a list decoding algorithm for the above codes. The algorithm follows the linear-algebraic list decoding algorithm for folded Reed-Solomon codes. Suppose a codeword \((3)\) encoding \( f \in \text{Im}(\kappa_{P_0}) \subseteq L((k+2g_e-1)P_{\infty}) \) is transmitted and received as

\[
y = \begin{pmatrix}
y_{1,1} & y_{1,2} & \cdots & y_{1,m} \\
y_{2,1} & y_{2,2} & \cdots \\
\vdots \\
y_{N,1} & \cdots & \cdots & y_{N,m}
\end{pmatrix}
\]

where some columns are erroneous. Let \( s > 1 \) be an integer parameter associated with the decoder.

**Lemma 3.5.** Given a received word \( y \) as in (4), using \( \text{poly}(N) \) operations over \( F_q \), we can find a nonzero linear polynomial in \( F_q[y_1, y_2, \ldots, y_N] \) of the form

\[
Q(y_1, y_2, \ldots, y_N) = A_0 + A_1 y_1 + A_2 y_2 + \cdots + A_N y_N
\]

satisfying

\[
Q(y_{i,j}, y_{i,j+1}, \cdots, y_{i,j+s-1}) = A_0(P_i^{s-1}) + A_1(P_i^{s-1})y_{i,j+1} + \cdots + A_s(P_i^{s-1})y_{i,j+s} = 0
\]

for \( i = 1, 2, \ldots, N \) and \( j = 0, 1, \ldots, m-s \). The coefficients \( A_i \) of \( Q \) satisfy \( A_i \in L(DP_{\infty}) \) for \( i = 1, 2, \ldots, s \) and \( A_0 \in L(D+k+2g_e-1)P_{\infty} \) for a “degree” parameter \( D \) chosen as

\[
D = \left\lfloor \frac{N(m-s+1)}{N(m-s+1) - k + (s-1)g_e + 1} \right\rfloor + 1.
\]

**Proof.** If we fix a basis of \( L(DP_{\infty}) \) (of the form (1)) and extend it to a basis of \( L((D+k+2g_e-1)P_{\infty}) \), then the number of freedoms of \( A_0 \) is at least \( D + k + g_e \) and the number of freedoms of \( A_i \) is at least \( D - g_e + 1 \) for \( i > 1 \). Thus, the total number of freedoms in the polynomial \( Q \) equals

\[
s(D - g_e + 1) + D + k + g_e = (s+1)(D+1) - (s-1)g_e + 1 + k > N(m-s+1).
\]
for the above choice (7) of $D$. The interpolation requirements on $Q \in \mathcal{E}[Y_1, \ldots, Y_s]$ are the following:

$$Q(y_{i,j}, y_{i,j+1}, \cdots, y_{i,j+s+1}) = A_0(P^{x_1}_t) + A_1(P^{x_2}_t)y_{i,j+1} + \cdots + A_s(P^{x_s}_t)y_{i,j+s} = 0$$

for $i = 1, 2, \ldots, N$ and $j = 0, 1, \ldots, m - s$. The interpolation requirements on $Q$ give a total of $N(m - s + 1)$ homogeneous linear equations that the coefficients of the $A_i$’s w.r.t. the chosen basis of $\mathcal{L}(D + k + 2g_e - 1)P_\infty$ must satisfy. Since the number of such coefficients (degrees of freedom in $Q$) exceeds $N(m - s + 1)$, we can conclude that such a linear polynomial $Q$ as required by the lemma must exist, and can be found by solving a homogeneous linear system over $\mathbb{F}_q$ with about $N(m - s + 1)$ variables and constraints.

Similar to earlier interpolation based list decoding algorithms, the following lemma gives an algebraic condition that the message functions $f \in \mathcal{L}(D + k + 2g_e - 1)P_\infty$ we are interested in list decoding must satisfy. The proof is a standard argument comparing the pole number to the order of zeros.

**Lemma 3.6.** If $f$ is a function in $\mathcal{L}(D + k + 2g_e - 1)P_\infty$ whose encoding (3) agrees with the received word $y$ in at least $t$ columns with $t > \frac{D + k + 2g_e - 1}{m - s + 1}$, then

$$Q(f, f^{s-1}, \ldots, f^{s-(i-1)}) = A_0 + A_1f + A_2f^{s-1} + \cdots + A_sf^{s-(i-1)} = 0$$

(10)

**Proof.** The proof proceeds by comparing the number of zeros of the function $Q(f, f^{s-1}, \ldots, f^{s-(i-1)}) = A_0 + A_1f + A_2f^{s-1} + \cdots + A_sf^{s-(i-1)}$ with $D + k + 2g_e - 1$. Note that $Q(f, f^{s-1}, \ldots, f^{s-(i-1)})$ is a function in $\mathcal{L}(D + k + 2g_e - 1)P_\infty$. If column $i$ of the encoding (3) of $f$ agrees with $y$, then for all $j = 0, 1, \ldots, m - s$, we have

$$0 = A_0(P^{x_1}_t) + A_1(P^{x_2}_t)y_{i,j+1} + A_2(P^{x_3}_t)y_{i,j+2} + \cdots + A_s(P^{x_s}_t)y_{i,j+s}$$

$$= A_0(P^{x_1}_t) + A_1(P^{x_2}_t)f(P^{x_3}_t) + A_2(P^{x_3}_t)f(P^{x_4}_t) + \cdots + A_s(P^{x_s}_t)f(P^{x_{s+1}}_t)$$

$$= A_0(P^{x_1}_t) + A_1(P^{x_2}_t)f(P^{x_3}_t) + A_2(P^{x_3}_t)f^{s-1}(P^{x_4}_t) + \cdots + A_s(P^{x_s}_t)f^{s-(i-1)}(P^{x_{s+1}}_t)$$

$$= (A_0 + A_1f + A_2f^{s-1} + \cdots + A_sf^{s-(i-1)})(P^{x_{s+1}}_t).$$

Note that here we use the fact that $f^s(P^{x_s}_t) = f(P^{x_s}_t)$, or equivalently $f^{s-1}(P^{x_s}_t) = f^{s-1}(P)$. In other words, $Q(f, f^{s-1}, \ldots, f^{s-(i-1)})$ has $(m - s + 1)$ distinct zeros from this agreeing column. Thus, there are a total of at least $t(m - s + 1)$ zeros for all the agreeing columns. Hence, $Q(f, f^{s-1}, \ldots, f^{s-(i-1)})$ must be the zero function when $t(m - s + 1) > D + k + 2g_e - 1$.

**Solving the functional equation for $f$.** Our goal next is to recover the list of solutions $f$ to the functional equation (10). Recall that our message functions lie in $\text{Im}(\kappa P_0)$, so we can recover $f$ by recovering the top $k$ coefficients $(f_0, f_1, \ldots, f_{k-1})$ of its local expansion $f = \sum_{j=0}^{k-1} f_j x^j$ at $P_0$. We now prove that $(f_0, f_1, \ldots, f_{k-1})$ for $f$ satisfying Equation (10) belong to a “periodic” subspace (in the sense of Definition 1) of not too large dimension.

**Lemma 3.7.** The set of solutions $(f_0, f_1, \ldots, f_{k-1}) \in \mathbb{F}_q^k$ such that $f = f_0 + f_1x + f_2x^2 + \cdots \in \mathcal{L}(k + 2g_e - 1)P_\infty$ obeys equation

$$A_0 + A_1f + A_2f^{s-1} + \cdots + A_sf^{s-(i-1)} = 0,$$

(11)

when the $A_i$’s obey the pole order restrictions of Lemma 3.5 and at least one $A_i$ is nonzero, is an affine subspace $W$ of dimension at most $(s - 1) \left\lceil \frac{k}{s} \right\rceil$.

Furthermore, there are at most $q^{Nm+1}$ possible choices of the subspace $W$ (as a function of the $A_i$’s), each of which is $(q - 1)$-periodic. Given the representation of each $A_i$ w.r.t. the basis (1), we can find a representation of $W$ in terms of the periodic subspace $U$ of dimension less than $s$, and the affine shifts in each window of $q - 1$ coordinates, in the sense of Definition 1.

**Proof.** Let $u = \min\{\nu_{P_0}(A_i) : i = 1, 2, \ldots, s\}$. Then it is clear that $u \geq 0$ and $\nu_{P_0}(A_0) \geq u$. Each $A_i$ has a local expansion at $P_0$:

$$A_i = x^n \sum_{j=0}^{\infty} a_{i,j} x^j$$

for $i = 0, 1, \ldots, s - 1$, which can be efficiently computed from the basis representation of the $A_i$’s. From the definition of $u$, one knows that the polynomial

$$B_0(X) := a_{1,0} + a_{2,0}X + \cdots + a_{s,0}X^{s-1}$$

is nonzero. Assume that at $P_0$, the function $f$ has a local expansion $\sum_{j=0}^{\infty} f_j x^j$. Then $f^{s-1}$ has a local expansion at $P_0$ as follows

$$f^{s-1} = \sum_{j=0}^{\infty} \xi^j f_j x^j,$$

where $\xi = 1/\gamma$. The coefficient of $x^{d+u}$ in the local expansion of $Q(f, f^{s-1}, \ldots, f^{s-(i-1)})$ is

$$0 = B_0(\xi^i) f_d + \sum_{j=0}^{d-1} b_{i,j} f_j + a_{0,d},$$

(12)

for $b_i \in \mathbb{F}_q$ is a linear combination of $a_{i,j}$ which does not involve $f_j$. Hence, $f_d$ is uniquely determined by $f_0, \ldots, f_{d-1}$ as long as $B_0(\xi^i) \neq 0$. Let $S := \{0 \leq i \leq q - 2 : B_0(\xi^i) = 0\}$. Then it is clear that $|S| \leq s - 1$ since the order of $\xi$ is $q - 1$ and $B_0(X)$ has degree at most $s - 1$. Thus, $B_0(\xi^i) \neq 0$ if and only if $j \mod (q - 1) \notin S$, and in this case $f_d$ is a fixed affine linear combination of $f_i$ for $0 \leq i < j$. Note that $B_0(X)$ has at most $(s - 1) \left\lceil \frac{k}{s} \right\rceil$ roots among $\{\xi^i : i = 0, 1, \ldots, k - 1\}$.

It follows that the set of solutions $(f_0, f_1, \ldots, f_{k-1})$ is an affine space $W \subset \mathbb{F}_q^k$, and the dimension of $W$ is at most $(s - 1) \left\lceil \frac{k}{s} \right\rceil$.

The fact that $W$ is $(q, q - 1)$-smooth follows from (12) and noting that the coefficients $b_{d-s,j}$ for $j \geq 1$ in that equation are given by $B_j(\xi^{i-s})$ where $B_j(X) := a_{s+1,j} + a_{s+2,j}X + \cdots + a_{s+s}X^{s-1}$. Therefore, once the values of $f_i, 0 \leq i \leq j < (q - 1)(q - 1) - 1$ are fixed, the possible choices for the next block of $(q - 1)$ coordinates $f_{j+1}(q-1), \ldots, f_{j+q-1}$ lie in an affine shift of a fixed subspace of dimension at most $(s - 1)$.

Further, this shift is an easily computed affine linear combination of the $f_i$’s in the previous blocks. This implies

344
the efficient computability of the claimed representation of $W$.

Finally, by the choice of $D$ in (7), the total number of possible $(A_0, A_1, \ldots, A_s)$ and hence the number of possible functional equations (11), is at most $q^{N(m-s+1)+s+1} \leq q^{Nm+s+1}$. Therefore, the number of possible candidate subspaces $W$ is also at most $q^{Nm+s+1}$. \hfill \square

Combining Lemmas 3.6 and 3.7, we conclude, after some simple calculations, that one can find a representation of the $(s, q-1)$-periodic subspace containing all candidate messages $(f_0, f_1, \ldots, f_{k-1})$ in polynomial time, when the fraction of errors $\tau = 1 - t/N$ satisfies

$$\tau \leq \frac{s}{s+1} - \frac{s}{s+1} \frac{k}{N(m-s+1)} - \frac{3m}{m-s+1} \frac{q_k}{mN}.$$  \hfill (13)

Pruning the subspace. Applying Lemma 3.7 directly we would get a list size bound of $q^{o(k^3)}$ which would be super-polynomial in the code length unless $k = O(q)$. Thus this idea does not directly allow us to get good list decodable codes while keeping the base field size small or achieve a list size that grows polynomially in $s$. Instead what we show is that by only encoding $(f_0, f_1, \ldots, f_{k-1}) \in \mathbb{F}_q^k$ that are restricted to belong to a special subspace-evasive set, we can (i) bring down the list size, and (ii) find this list efficiently in polynomial time (and further the exponent of the polynomial is independent of $\varepsilon$, the gap to capacity). To this end, we develop the necessary machinery concerning subspace evasive sets next. Later, in Section 5, we combine these subspace evasive sets with our folded Hermitian codes to get good list-decodable codes.

4. SUBSPACE EVASIVE SETS WITH ADDITIONAL STRUCTURE

Let us first recall the notion of “ordinary” subspace-evasive sets from [6].

**Definition 4.** A subset $S \subseteq \mathbb{F}_q^k$ is said to be $(d, \ell)$-subspace-evasive if for all $d$-dimensional affine subspaces $W$ of $\mathbb{F}_q^k$, we have $|S \cap W| \leq \ell$.

Next we define the notion of easiness w.r.t a collection of subspaces instead of all subspaces of a particular dimension.

**Definition 5.** Let $\mathcal{F}$ be a family of (affine) subspaces of $\mathbb{F}_q^k$, each of dimension at most $d$. A subset $S \subseteq \mathbb{F}_q^k$ is said to be $(\mathcal{F}, d, \ell)$-evasive if for all $W \in \mathcal{F}$, we have $|S \cap W| \leq \ell$.

The key to pruning the list to a small size is the notion of a hierarchical subspace-evasive set, which is defined as a subset of $\mathbb{F}_q^k$ with the property that some of its prefixes are subspace-evasive with respect to $(\Delta, d)$-periodic subspaces. We will show how the special subspace-evasive sets help towards pruning the list in our list decoding context in Lemma 4.2.

**Definition 6.** Let $\mathcal{F}$ be a family of $(\Delta, d)$-periodic subspaces of $\mathbb{F}_q^k$. A subset $S \subseteq \mathbb{F}_q^k$ is said to be $(\mathcal{F}, s, \Delta, L)$-h.s.e (for hierarchically subspace evasive for block size $\Delta$) if for every affine subspace $W \in \mathcal{F}$, the following bound holds for $j = 1, 2, \ldots, b$: $|\text{proj}_j^\Delta(S) \cap \text{proj}_j^\Delta(W)| \leq L$.

Pseudorandom construction of large h.s.e. subsets. Our goal is to give a randomized construction of large h.s.e sets that works with high probability, with the further properties that one can index into elements of this set efficiently (necessary for efficient encoding), and one can check membership in the set efficiently (which is important for efficient decoding).

Suppose, for some fixed subset $\mathcal{F}$ of $(\Delta, d)$-periodic subspaces of $\mathbb{F}_q^k$, we are interested in an $(\mathcal{F}, s, \Delta, L)$-h.s.e subset of $\mathbb{F}_q^k$ of size $\approx q^{(1-\zeta)k}$ for a constant $\zeta$, $1/\Delta < \zeta < 1$. For simplicity, let us assume that the block size $\Delta$ divides $k$, though arbitrary $k$ can be easily handled. Define $b = \frac{k}{\Delta}$ to be the number of blocks. The parameters $b, \Delta, k$ and field size $q$ will be considered fixed for the rest of the discussion in this section.

Our construction will use some arbitrary fixed subsets $\Lambda_1, \Lambda_2, \ldots, \Lambda_b$ where $\Lambda_i \subseteq \mathbb{F}_q^i\Delta$ with $|\Lambda_i| = q^{i(1-\zeta)\Delta}$.

Note that by virtue of the $\zeta$-wise independence of the values of a random degree $\Delta$ polynomial, and the independent choices of $\mathcal{P}_i, \mathcal{Q}_i$, for a collection of subsets $T_i \subseteq \mathbb{F}_q^\Delta$ with $|T_i| \leq \ell$, the values $\{\mathcal{P}_i(\alpha_i), \mathcal{Q}_i(\alpha_i)\}_{\alpha_i \in T_i}$ are independent random values in $\mathbb{F}_q^\Delta$, and moreover the values for different $i$’s are independent.

With these ingredients in place, we now describe our construction of the h.s.e subset. Our construction is based on an encoding map from strings in $\mathbb{F}_q^{(1-3\zeta)k}$ to $\mathbb{F}_q^k$, and letting the image of this map be the set. (The full version of the paper gradually works its way to this construction, but here we directly give this construction.)

**Definition 7 (Encoding into the h.s.e set).** Given the polynomials $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_b$ and $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_b$, and the subsets $\Lambda_1 \subseteq \mathbb{F}_q^\Delta$, the encoding of $x = (x_1, x_2, \ldots, x_b)$ where $x_i \in \mathbb{F}_q^{(1-3\zeta)\Delta}$, proceeds as follows:

For $i = 1, 2, \ldots, b$:

- Let $\beta_i \in \mathbb{F}_q^{\Delta}\lambda$ be the lexicographically first string such that $\mathcal{P}_i(\rho_i(x_1 \circ \beta_1 \circ \cdots \circ x_i \circ \beta_i)) \in \Lambda_i$ and $\mathcal{Q}_i(\rho_i(x_1 \circ \beta_1 \circ \cdots \circ x_i \circ \beta_i)) \in \Lambda_i$. If no such $\beta_i$ exists, fail.

Output $x_1 \circ \beta_1 \circ x_2 \circ \beta_2 \circ \cdots \circ x_b \circ \beta_b \in \mathbb{F}_q^k$ as the encoding of $x$. We will denote the above encoding map by HSE and refer to $\Delta$ as its period size (we suppress the dependence of the map on the $\mathcal{P}$’s and $\mathcal{Q}$’s for convenience).

The following is the main theorem regarding this construction. The proof that HSE is well defined is based on the fact at each stage $i$, regardless of the choice of $\beta_1, \ldots, \beta_{i-1}$, there...
are an expected $q^{-\Delta}$ choices of $\beta_i$ satisfying the required condition, and this random variable is well-concentrated due to $\lambda$-wise independence. The h.s.e property is proved by a careful inductive argument, showing that for $i = 1, 2, \ldots, b$, the projection of the image HSE onto the first $\Delta$ coordinates intersects the corresponding projection of all subspaces in $\mathcal{F}$ in at most $L$ points. By the definition of $(s, \Delta)$-periodic subspaces, this means that at the next stage $i + 1$, only $L \cdot q^s$ candidates are in play to belong to the intersection, which is small enough for us to afford a union bound and perform the inductive step (the $\lambda$-wise independence is again used in this step). The details appear in the full version of the paper.

**Theorem 4.1.** (Main construction of h.s.e. subsets) Suppose $b, c, \Delta, k, s$ are positive integers and $\zeta \in (0, 1)$ such that following requirements are met:

$$k = b \Delta; \quad s < \zeta \Delta / 10; \quad q^{\Delta} \geq (2cq^{b})^{10}/9. \quad (14)$$

Let $\mathcal{F}$ be a family of $(s, \Delta)$-periodic subspaces of $\mathbb{F}_q^k$ with $|\mathcal{F}| \leq q^{b}$. Then, for a random and independent choice of polynomials $P, Q_i \in \mathbb{F}_q[x]$ of degree $\lambda = \max\{6b, k + 1\}$ and any subsets $\Lambda_i$ of size $q^{(s-1)\Delta}$ for $i = 1, 2, \ldots, b$, the following conditions both hold with probability at least $1 - 2^{-10(b)}$.

1. HSE : $\mathbb{F}_q^{1-3\zeta k} \rightarrow \mathbb{F}_q^k$ from Definition 7 is a well-defined injective map, and can be computed using $O(q^{\zeta \Delta \log k})$ operations over $\mathbb{F}_q$.

2. The set $H = \Gamma(P_1, \ldots, P_b) \cap \Gamma(Q_1, \ldots, Q_b)$, and in particular the image of HSE, is a $(\mathcal{F}, s, \Delta, c\cdot k)$-h.s.e and a $(\mathcal{F}, b, 20c/\zeta)$-evasive subset of $\mathbb{F}_q^k$.

**Efficient computation of intersection with h.s.e. subsets.** The key aspect which makes h.s.e subsets useful in our context to prune the affine space of candidate messages, is the following claim which shows that intersection of a $(s, \Delta)$-periodic subspace with our h.s.e set can found efficiently.

**Lemma 4.2.** Suppose polynomials $P_i, Q_i, i = 1, 2, \ldots, b$, of degree $\lambda = \max\{6b, k + 1\}$ have been picked so that the map HSE satisfies the conditions of Theorem 4.1 w.r.t some family $\mathcal{F}$ of at most $q^b$ affine subspaces of $\mathbb{F}_q^k$ each of which is $(s, \Delta)$-periodic. Then given a representation of $W$ in $\mathcal{F}$ (as in Definition 1), we can find the list of at most $O(c/\zeta)$ values of $x \in \mathbb{F}_q^{1-3\zeta k}$ such that $\text{HSE}(x) \in \mathcal{F}$ using $O(c^2(qk^2 + q^{3\Delta \log k})^s \log k)$ operations over $\mathbb{F}_q$.

**Proof**. The fact that there are at most $\ell$ solutions $x$ follows immediately from the fact that the image of HSE is $(\mathcal{F}, b, \ell)$-evasive. So we only need to argue about the time complexity.

For $1 \leq i \leq b$, define $H_i = \text{proj}_{\Delta}(H)$ where $H = \Gamma(P_1, \ldots, P_b) \cap \Gamma(Q_1, \ldots, Q_b)$. Likewise, define $W_i = \text{proj}_{\Delta}(W)$. To compute the intersection $H \cap W$ list efficiently, we iteratively find $W_i \cap H_i$ for $i = 1, 2, \ldots, b$ as follows. Recall that we know that $|H_i \cap W_i| \leq L$ for each $i$ as $H$ is $(\mathcal{F}, s, \Delta, \lambda)$-h.s.e. For each of the at most $L$ $s$-complete candidates in $W_i \cap H_i$, as $W$ is $(s, \Delta)$-periodic, there are at most $q^s$ possible extensions to the next block of $(q - 1)$ coordinates which we can find and list using $O(q^s \cdot \Delta)$ operations. (The $k\Delta$ term comes from computing the affine shift for the $i$th block for that particular prefix of $(i - 1)\Delta$ symbols.)

Then we test each of $L \cdot q^s$ candidates for membership in $\Gamma$, which can be done using $O(ck^2 \log k)$ $\mathbb{F}_q$-operations time by evaluating the degree $\lambda$ polynomial and checking that the resulting value belongs to $\Lambda$. By the $(\mathcal{F}, s, \Delta, \lambda)$-h.s.e property of $H$ there are at most $L$ of these candidates that can belong to $H_i$, thus bringing our list size back to $L$. The runtime for each iterative step is $O(Lq^s k^2)$ $\mathbb{F}_q$-operations, leading to an overall runtime for all $b < k$ stages of $O(Lq^s k^2 \log k)$ operations over $\mathbb{F}_q$ to recover the intersection $H \cap W$. Finally for each $y \in H \cap W$, we can check if it is the range of HSE by writing $y = x_1 \circ \beta_1 \circ \cdots \circ x_b \circ \beta_\lambda$ and checking that $\text{HSE}(x_1 \circ x_2 \circ \cdots \circ x_b) = y$, which takes $O(q^s k^2 \log k)$ operations over $\mathbb{F}_q$. □

5. COMBINING FOLDED HERMITIAN CODES AND H.S.E SETS

Instead of encoding arbitrary $f \in \mathbb{F}_q^n$ by the folded Hermitian code (Definition 3), we can restrict the messages $f$ to belong to the range of our h.s.e set, so that the affine space of solutions guaranteed by Lemma 3.7 can be efficiently pruned to a small list. The formal claim is below.

**Theorem 5.1.** Let $c \geq 2$ be an integer, $r \geq 2c$ be a large enough prime power, $q = r^2$, and $\zeta \in (1/q, 1)$. Let $k \leq q^c/2$ be a positive integer. Let $s, m$ be positive integers satisfying $1 \leq s \leq \min \{q - 1, \zeta q/2\}$. Finally let $N$ be an integer satisfying $k + 2er^c \leq Nm \leq (q - 1)r^c$.

Consider the code $C$ with encoding $E_1 : \mathbb{F}_q^{1-3\zeta k} \rightarrow (\mathbb{F}_q^m)^N$ defined as

$$E_1(x) = FH(N, k, q, e, m)(\text{HSE}(x)),$$

for HSE : $\mathbb{F}_q^{1-3\zeta k} \rightarrow \mathbb{F}_q^k$ from Definition 7 for a period size $\Delta = q - 1$.

Then, with high probability over the choice of HSE, this code has rate $R = (1 - 3\zeta)k/(Nm)$, can be encoded in poly$(Nm^{q^c})$ time, and is $(\tau, \ell)$-list decodable in time poly$(Nm^{q^c})$ for $\ell \leq O(1/(R\zeta))$ and

$$\tau = s + 1 \left(1 - \frac{k}{N(m - s + 1)}\right) - \frac{3m}{m - s + 1} \frac{er^c}{N m}.$$

**Proof**. This follows by just combining the ingredients we have developed so far. Since $g_s \leq er^c$ by (2), the condition on $N_m$ meets the requirement for the construction of the folded Hermitian tower based code in Definition 2.

Wip, the map HSE is well-defined and injective, and so $E_1$ is an injective encoding. The rate of the code is therefore clearly as claimed. With $\Delta = q - 1$, one can check that the conditions of Theorem 4.1 are met for our choice of $\zeta, s, q, k$. By Theorem 4.1, Part 1, HSE can be computed in time poly$(Nm^{q^c})$ and hence so can $E_1$ (as FH is efficiently encodable as well).

The claimed value of the error fraction $\tau$ satisfies (13) since the genus is at most $er^c$ by (2). By Lemma 3.7, we know that the candidate messages found by the decoder lie in one of at most $q^{\zeta N_m}$ possible $(s, q - 1)$-periodic subspaces. Appealing to Theorem 4.1 and Lemma 4.2 with the choice $c = 2N_m/k = O(1/R)$, we conclude that there is a decoding algorithm running in time poly$(Nm^{q^c})$ to list decode $C$ from a fraction $\tau$ of errors, outputting at most $O(1/(R\zeta))$ messages in the worst-case. □
Let \( \varepsilon > 0 \) be a small positive constant, and a family of codes of length \( N \) (assumed large enough) and rate \( R \in (0,1) \) is sought. Pick \( n \) to be a growing parameter.

By picking \( s = \Theta(1/\varepsilon) \), \( \Theta = \Theta(1/\varepsilon^2) \), \( r = \lceil \log n \rceil \), \( e = \left\lfloor \frac{\log n}{\log \log n} \right\rfloor \), \( \zeta = \log n / \log \log n \), \( N = \left\lfloor \frac{2^{n-R}}{\varepsilon m} \right\rfloor \), and \( k \) proportional to \( Nm \) in Theorem 5.1, we can conclude the following.

**Corollary 5.2.** For any \( R \in (0,1) \) and positive constant \( \varepsilon \in (0,1) \), there is a Monte Carlo construction of a family of \( \Theta(1/\varepsilon^2) \)-list decodable and \( (1-R-\varepsilon, O(R^{-1} \log N \log \log N)) \)-list decodable in \( \text{poly}(N, 1/\varepsilon) \) time, where \( N \) is the block length of the code.

Our main result (Theorem 1.1) achieves better parameters than the above — an alphabet size of \( \exp(O(1/\varepsilon^2)) \) and list-size of \( O(1/(R\varepsilon)) \). This is based on the Garcia-Stichtenoth tower and is described next.

### 6. FOLDED CODES FROM THE GARCIA-STICHENTOHN TOWER

Compared with the Hermitian tower of function fields, the Garcia-Stichtenoth tower of function fields yields folded codes with better parameters due to the fact that it achieves the optimal ratio between number of rational places and the genus. The construction of folded codes from the Garcia-Stichtenoth tower is almost identical to the one from the Hermitian tower except for one major difference: the redefined code from the Garcia-Stichtenoth tower is constructed in terms of the local expansion at point \( P_\infty \), while in the Hermitian case local expansion at \( P_0 \) is considered. For convenience of the reader, we give a parallel description of folded codes from the Garcia-Stichtenoth tower, while only sketching the identical parts.

#### 6.1 Background on Garcia-Stichtenoth tower

Again let \( r \) be a prime power and let \( q = r^2 \). We denote by \( \mathbb{F}_q \) the finite field with \( q \) elements. The Garcia-Stichtenoth towers that we are going to use for our code construction were discussed in [3, 4]. The reader may refer to [3, 4] for the detailed background on the Garcia-Stichtenoth function tower. There are two optimal Garcia-Stichtenoth towers that are equivalent. For simplicity, we introduce the tower defined by the following recursive equations [4]

\[
x_i^{e+1} + x_i = \frac{x_i^e}{x_i^{e-1} + 1}, \quad i = 1, 2, \ldots, e - 1.
\]

Put \( K_e = \mathbb{F}_q(x_1, x_2, \ldots, x_e) \) for \( e \geq 2 \).

**Rational places.** The function field \( K_e \) has at least \( r^{e-1}(r^2 - r) + 1 \) rational places. One of these is the “point at infinity” which is the unique pole \( P_\infty \) of \( x_1 \) (and is fully ramified). The other \( r^{e-1}(r^2 - r) \) come from the rational places lying over the unique zero of \( x_1 - \alpha \) for each \( \alpha \in \mathbb{F}_q \) with \( \alpha^e + \alpha \neq 0 \). Note that for every \( \alpha \in \mathbb{F}_q \) with \( \alpha^e + \alpha \neq 0 \), the unique zero of \( x_1 - \alpha \) splits completely in \( K_e \), i.e., there are \( r^{e-1} \) rational places lying over the zero of \( x_1 - \alpha \). Let \( \mathbb{P} \) be the set of all the rational places lying over the zero of \( x_1 - \alpha \) for all \( \alpha \in \mathbb{F}_q \) with \( \alpha^e + \alpha \neq 0 \). Then, intuitively, one can think of the \( r^{e-1}(r^2 - r) \) rational places in \( \mathbb{P} \) as being given by \( e \)-tuples \((\alpha_1, \alpha_2, \ldots, \alpha_e) \in \mathbb{F}_q^e \) that satisfy \( \alpha_{i+1}^{e+1} + \alpha_{i+1} = \frac{\alpha_i^e}{\alpha_i^{e-1} + 1} \) for \( i = 1, 2, \ldots, e - 1 \) and \( \alpha_1^e + \alpha_1 \neq 0 \). For each value of \( \alpha \in \mathbb{F}_q \), there are precisely \( r \) solutions to \( \beta \in \mathbb{F}_q \) satisfying \( \beta^e + \beta = \frac{\alpha^e}{\alpha^{e-1} + 1} \), so the number of such \( e \)-tuples is \( r^{e-1}(r^2 - r) \) \( (r^2 - r) \) choices for \( \alpha_1 \), and then \( r \) choices for each successive \( \alpha_i, 2 \leq i \leq e \).

**Riemann-Roch spaces.** As shown in [13], every function of \( K_e \) with a pole only at \( P_\infty \) has an expression of the form

\[
x_i^e \left( \sum_{i=0}^{r-1} \prod_{i=0}^{r-1} c_{i_1} x_1^{e_1} \cdots x_e^{e_e} \right)
\]

where \( \alpha \geq 0, c_i \in \mathbb{F}_q \), and for \( 1 \leq j < e, c_{i_j} = x_{i_j}^{r-1} + 1 \) and \( c_1 = h_1h_2 \cdots h_j \). Moreover, Shum et al. [13] present an algorithm running in time polynomial in \( l \) that outputs a basis of over \( \mathbb{F}_q \) of \( \mathcal{L}(P_\infty) \) explicitly in the above form.

We stress that evaluating elements of \( \mathcal{L}(P_\infty) \) at the rational places of \( \mathbb{P} \) is easy: we simply have to evaluate a linear combination of the monomials allowed in (15) at the tuples \((\alpha_1, \alpha_2, \ldots, \alpha_e) \in \mathbb{F}_q^e \) mentioned above. In other words, it is just evaluating an \( e \)-variate polynomial at a specific subset of \( r^{e-1}(r^2 - r) \) points of \( \mathbb{F}_q \), and can be accomplished in polynomial time.

**Genus.** The genus \( g_e \) of the function field \( K_e \) is given by

\[
ge_e = \begin{cases} \frac{(r^{e/2} - 1)^2}{2} & \text{if } e \text{ is even} \\ \frac{(r^{(e+1)/2} - 1)^2}{2} & \text{if } e \text{ is odd}. \end{cases}
\]

Thus the genus \( g_e \) is at most \( r^e \). (Compare this with the \( er^e \) bound for the Hermitian tower; this smaller genus is what allows to pick \( e \) as large as we want in the Garcia-Stichtenoth tower, while keeping the field size \( q \) fixed.)

**A useful automorphism.** Let \( \gamma \) be a primitive element of \( \mathbb{P} \), and consider the automorphism \( \sigma \in \text{Aut}(K_e/\mathbb{F}_q) \) defined by

\[
\sigma: x_i \mapsto \gamma^{(i+1)r^{i-1}} x_i \quad \text{for } i = 1, 2, \ldots, e.
\]

Then the order of \( \sigma \) is \( r - 1 \) and furthermore, we have the following facts:

(i) \( \sigma \) keeps \( P_\infty \) unchanged, i.e., \( \sigma(P_\infty) = P_\infty \);

(ii) Let \( \mathbb{P} \) be the set of all the rational places lying over \( x_1 - \alpha \) for all \( \alpha \in \mathbb{F}_q \) with \( \alpha^e + \alpha \neq 0 \). Then \( |\mathbb{P}| = (r-1)r^e \). Moreover, \( \sigma \) divides \( \mathbb{P} \) into \( r^e \) orbits and each orbit has \( r - 1 \) places. For an integer \( m \) with \( 1 \leq m \leq r - 1 \), we can label \( N \) \( m \) distinct elements

\[
P_1, P_2^m, \ldots, P_t^m, \ldots, P_N, P_N^m, \ldots, P_N^{m-1}
\]

in \( \mathbb{P} \), as long as \( N \leq r^e \left( \frac{2^m}{m} \right) \).

The folded codes from the Garcia-Stichtenoth tower are defined similarly to the Hermitian case.

**Definition 8. (Folded codes from the GS tower)** Assume that \( m, k \) are positive integers satisfying \( 1 \leq m < r - 1 \) and \( l/m < N \leq r^e \left( \frac{2^m}{m} \right) \). The folded code from \( K_e \) with parameters \( N, l, q, e, m \), denoted by \( \text{FGS}(N, l, q, e, m) \), encodes a message function \( f \in \mathcal{L}(P_\infty) \) as

\[
f \mapsto \left( \begin{array}{c} f(P_1) \\ f(P_2) \\ \vdots \\ f(P_t^m) \\ \vdots \\ f(P_N^{m-1}) \end{array} \right) \in \mathbb{F}_q^N.
\]

(16)
Then we have a similar result on parameters of $\overline{F}_{q}(N, l, q, e, m)$.

**Lemma 6.1.** The above code $\overline{F}_{q}(N, l, q, e, m)$ is an $\mathbb{F}_{q}$-linear code over alphabet size $q^{n}$, rate at least $\frac{e}{m}+\frac{1}{m}$, and minimum distance at least $N - \frac{q}{2}$.

### 6.2 Redefining the code in terms of local expansion at $P_{\infty}$

In the Hermitian case, we use coefficients of its power series expansion around $P_{0}$. However, for the Garcia-Stichtenoth tower we do not have such a nice point $P_{0}$. Fortunately, we can use point $P_{\infty}$ to achieve our mission.

Again for our decoding, we will actually recover the message $f \in \mathcal{L}(P_{\infty})$ in terms of the coefficients of its power series expansion around $P_{\infty}$

$$f = T^{-i}(f_{0} + f_{1}T + f_{2}T^{2} + \cdots)$$

where $T := \frac{1}{x}$ is the local parameter at $P_{\infty}$ (which means that $x_{e}$ has exactly one pole at $P_{\infty}$, i.e., $v_{P_{\infty}}(x_{e}) = -1$).

In this case we can also show that one can efficiently move back-and-forth between the representation of $f \in \mathcal{L}(P_{\infty})$ in terms of a basis for $\mathcal{L}(P_{\infty})$ and its power series representation $(f_{0}, f_{1}, \ldots)$ around $P_{\infty}$. Since the mapping $f \mapsto (f_{0}, f_{1}, \ldots)$ is $\mathbb{F}_{q}$-linear, it suffices to compute the local expansion of $P_{\infty}$ as a basis for $\mathcal{L}(P_{\infty})$.

**Lemma 6.2.** For any $n$, one can compute the first $n$ terms of the local expansion of the basis elements $(15)$ at $P_{\infty}$ using poly$(n)$ operations over $\mathbb{F}_{q}$.

**Proof.** First let $h$ be a nonzero function in $\mathbb{F}_{q}(x_{1}, x_{2}, \ldots, x_{l})$ with $v_{P_{\infty}}(h) = v \in \mathbb{Z}$. Assume that the local expansion

$$h = T^{v}\sum_{j=0}^{\infty} a_{j}T^{j}$$

is known. To find the local expansion

$$1 = T^{-v}\sum_{j=0}^{\infty} c_{j}T^{j}$$

Consider the identity

$$1 = \left(\sum_{j=0}^{\infty} a_{j}T^{j}\right)\left(\sum_{j=0}^{\infty} c_{j}T^{j}\right)$$

Then by comparing the coefficients of $T^{i}$ in the above identity, one has $c_{0} = a_{0}$ and $c_{i} = -a_{0}^{-1}(c_{i-1}a_{1} + \cdots + c_{0}a_{i})$ can be easily computed recursively for all $i \geq 1$.

Thus, by the structure of the basis functions in $(15)$, it is sufficient to find an algorithm of efficiently finding local expansions of $x_{i}$ at $P_{\infty}$ for every $i = 1, 2, \ldots, e$. We can inductively find the local expansions of $x_{i}$ at $P_{\infty}$ as follows.

We note that $v_{P_{\infty}}(x_{i}) = -r^{i-1}$ for $i = 1, 2, \ldots, e$.

For $i = e$, $x_{e}$ has the local expansion $\frac{1}{x_{e}}$ at $P_{\infty}$.

Now assume that we know the local expansion of $x_{i}$. Then we can easily compute the local expansion of $x_{i}^{r}$ and hence the local expansion of $1/(x_{i} + 1)$. Let us assume that $1/(x_{i}^{r} + x_{i})$ has local expansion

$$1/(x_{i}^{r} + x_{i}) = T^{e-i+1}\sum_{j=0}^{\infty} a_{j}T^{j}$$

at $P_{\infty}$ for some $a_{j} \in \mathbb{F}_{q}$. Assume that $1/x_{i+1}$ has the local expansion $1/x_{i+1} = T^{e-i+1}\sum_{j=0}^{\infty} \beta_{j}T^{j}$. To find $\beta_{j}$, we consider the identity

$$T^{e-i+1}\sum_{j=0}^{\infty} \beta_{j}T^{j} + T^{e-i+2}\sum_{j=0}^{\infty} \beta_{j}T^{j} = \frac{1}{x_{i+1}} + \left(\frac{1}{x_{i+1}}\right)^{r}$$

However, for the Garcia-Stichtenoth code $\mathcal{P}_{\infty}$, we can therefore pick a pre-image in $\mathcal{L}(l = 2g_{e} - 1, P_{\infty})$. For convenience, we will denote an injective map making such a unique choice by $\kappa_{\mathcal{P}_{\infty}} : \mathcal{P}_{\infty} \rightarrow \mathcal{L}(l = 2g_{e} - 1, P_{\infty})$. By picking the pre-images of a basis of $\mathcal{P}_{\infty}$ and extending it by linearity, we can assume $\kappa_{\mathcal{P}_{\infty}}$ to be $\mathbb{F}_{q}$-linear, and thus specify it by a $(k + g_{e}) \times k$ matrix. We record this fact for easy reference below.

**Claim 6.3.** $\mathcal{E}_{\mathcal{P}_{\infty}}$ is an $\mathbb{F}_{q}$-linear surjective map. Further, we can compute $\mathcal{E}_{\mathcal{P}_{\infty}}$ using poly$(k, g_{e})$ operations over $\mathbb{F}_{q}$ given a representation of the input $f \in \mathcal{L}(l = 2g_{e} - 1, P_{\infty})$ in terms of the basis $(15)$.

The proof of this claim is similar to Claim 3.3. Note that the kernel of $\mathcal{E}_{\mathcal{P}_{\infty}}$ is $\mathcal{L}(l = 2g_{e} - 1, P_{\infty})$ which has dimension exactly $g_{e}$ by the Riemann-Roch theorem.

For each $(f_{0}, f_{1}, \ldots, f_{k-1}) \in \mathcal{P}_{\infty}$, we can therefore pick a pre-image in $\mathcal{L}(l = 2g_{e} - 1, P_{\infty})$. For convenience, we will denote an injective map making such a unique choice by $\kappa_{\mathcal{P}_{\infty}} : \mathcal{P}_{\infty} \rightarrow \mathcal{L}(l = 2g_{e} - 1, P_{\infty})$. By picking the pre-images of a basis of $\mathcal{P}_{\infty}$ and extending it by linearity, we can assume $\kappa_{\mathcal{P}_{\infty}}$ to be $\mathbb{F}_{q}$-linear, and thus specify it by a $(k + g_{e}) \times k$ matrix. We record this fact for easy reference below.

**Claim 6.4.** The map $\kappa_{\mathcal{P}_{\infty}} : \mathcal{P}_{\infty} \rightarrow \mathcal{L}(l = 2g_{e} - 1, P_{\infty})$ is $\mathbb{F}_{q}$-linear and injective. We can compute a representation of this linear transformation using poly$(k, g_{e})$ operations over $\mathbb{F}_{q}$, and the map itself can be evaluated using poly$(k, g_{e})$ operations over $\mathbb{F}_{q}$.

Now we redefine a version of the folded Garcia-Stichtenoth code that maps $\mathcal{P}_{\infty}$ to $(\mathbb{F}_{q}^{m})^{N}$ by composing the folded encoding $(16)$ from the original Definition 8 with $\kappa_{\mathcal{P}_{\infty}}$.

**Definition 9. (Folded Garcia-Stichtenoth code using local expansion)** The folded Garcia-Stichtenoth code ($\mathcal{F}_{GS}$ code for short) $\mathcal{F}_{GS}(N, k, q, e, m)$ maps $f = (f_{0}, f_{1}, \ldots, f_{k-1}) \in \mathcal{P}_{\infty}$ to $\mathcal{F}_{GS}(N, k = 2g_{e} - 1, q, e, m)\mathcal{E}_{\mathcal{P}_{\infty}}(f) \in (\mathbb{F}_{q}^{m})^{N}$.

The rate of the above code equals $k/(Nm)$ and its distance is at least $N - (k + 2g_{e} - 1)/m$. 

348
6.3 List decoding FGS codes

The list decoding part for the codes from the Garcia-Stichtenoth tower is almost identical to the Hermitian tower. We only sketch this part briefly.

If \( f \) is a function in \( \mathcal{L}(k + 2g_k - 1)P_{\infty} \) whose encoding (16) agrees with the received word \( y \) in at least \( t \) columns with \( t > \frac{k}{m - s + 1} + 1 \) and

\[
D = \left[ \frac{N(m - s + 1) - k + (s - 1)g_k + 1}{s + 1} \right],
\]

then there exist \( A_i \) in \( \mathcal{L}(DP_{\infty}) \) for \( i = 1, 2, \ldots, s \) and \( A_0 \) in \( \mathcal{L}((D + k + 2g_k - 1)P_{\infty}) \) such that they are not all zero and

\[
Q(f, f^{s^{-1}}, \ldots, f^{(s-1)^{-1}}) = A_0 + A_1 f + A_2 f^{s^{-1}} + \cdots + A_{s} f^{(s-1)^{-1}} = 0.
\]

(17)

Solving the functional equation for \( f \). As in the Hermitian case, our goal now is to recover the list of solutions \( f \) to the functional equation (17). Recall that our message functions lie in \( \text{Im}(K_{P_{\infty}}) \), so we can recover \( f \) by recovering the top \( k \) coefficients of \((f_0, f_1, \ldots, f_{k-1})\) of its local expansion.

\[
f = T^{-(k+2g_k-1)} \sum_{j=0}^{\infty} f_j T^j
\]

at \( P_{\infty} \). We now prove that \((f_0, f_1, \ldots, f_{k-1})\) for \( f \) satisfying Equation (17) belong to a “sparse” subspace of the form of Definition 1) of not too large dimension. The proof, which is similar to Lemma 3.7, is omitted here and can be found in the full version.

Lemma 6.5. The set of solutions \((f_0, f_1, \ldots, f_{k-1})\) in \( P_{\infty}^k \) such that \( f = T^{-(k+2g_k-1)} \sum_{j=0}^{\infty} f_j T^j \in \mathcal{L}((k + 2g_k - 1)P_{\infty}) \) obeys equation

\[
A_0 + A_1 f + A_2 f^{s^{-1}} + \cdots + A_{s} f^{(s-1)^{-1}} = 0
\]

when at least one \( A_i \) is nonzero is an affine subspace \( W \) of dimension at most \((s - 1) \left\lfloor \frac{k}{s - 1} \right\rfloor \).

Further, there are at most \( q^{s \left\lfloor \frac{k m + s + 1}{s - 1} \right\rfloor} \) possible choices of the subspace \( W \), each of which is \((s, r-1)-\)periodic.

Given the representation of each \( A_i \) w.r.t. the basis (15), we can find a representation of \( W \) in terms of the periodic subspace \( V \) of dimension less than \( s \), and the affine shifts in each window of \( r-1 \) coordinates, in the sense of Definition 1.

Similar to the bound (13) for the Hermitian case, we conclude, after some simple calculations and using the upper bound on \( g_k \leq r^s \), that one can find a representation of the \((s, r-1)\)-periodic subspace containing all candidate messages \((f_0, f_1, \ldots, f_{k-1})\) in polynomial time, when the fraction of errors \( \tau = 1 - \frac{1}{t} \) satisfies

\[
\tau \leq \frac{s}{s + 1} \left( 1 - \frac{k}{N(m - s + 1)} \right) - \frac{3m}{m - s + 1} \frac{r^s}{mN}.
\]

(20)

6.4 Combining FGS codes and h.s.e sets

Similarly to Section 5, we now show how to pre-code the messages of the FGS code with a h.s.e subset. The approach is similar, though we need one idea to ensure that we can pick parameters so that the base field \( P_q \) can be constant-sized and obtain a final list-size bound that is a constant independent of the code length. This idea is to work with a larger “period size” \( \Delta \) for the periodic subspaces, based on the following observation.

Observation 6.6. Let \( W \) be an \((s, \Delta)\)-periodic subspace of \( P_q^k \) for \( k = b \Delta \). Then \( W \) is also \((su, \Delta u)\)-periodic for every integer \( u \), \( 1 \leq u \leq b \).

As in the Hermitian case, instead of encoding arbitrary \( f \in P_q^k \) by the folded Garcia-Stichtenoth code (Definition 3), we will restrict the messages \( f \) to belong to the range of our h.s.e set. This will ensure that the affine space of solutions guaranteed by Lemma 6.5 can be efficiently pruned to a small list.

Theorem 6.7. Let \( r \) be a prime power, \( q = r^2 \), and \( e \geq 2 \) be an integer, and \( \zeta \in (0, 1) \). Let \( k < q^{\frac{\Delta}{2}} \) be a positive integer. Let \( \Delta \leq k \) be a multiple of \((r-1) \), say \( \Delta = u(r-1) \) for a positive integer \( u \).

Let \( s, m \) be positive integers satisfying \( 1 \leq s \leq m \leq r - 1 \) and \( s < \frac{\zeta r}{12} \). Finally let \( N \) be an integer satisfying \( k + 2r^e \leq N \tau \leq (r^{s(1)} - r^e)\).

Consider the code \( C_2 \) with encoding \( E_2 : P_q^{(1-3)k} \to (P_q^m)^N \) defined as

\[
E_2(x) = \text{FGS}(N, k, q, e, m)(HSE(x)),
\]

for \( HSE : P_q^{(1-3)k} \to P_q^m \) from Definition 7 for a period size \( \Delta \).

Then, with high probability over the choice of \( HSE \) with period size \( \Delta \), this code has rate \( R = (1-3)k/(Nm) \) can be encoded in \( \text{poly}(Nm^2r^\Delta) \) time, and is \((s, \ell)\)-list decodable in time \( \text{poly}(Nm^2r^\Delta) \) for \( \ell \leq O(1/(R\zeta)) \) and

\[
\tau = \frac{s}{s + 1} \left( 1 - \frac{k}{N(m - s + 1)} \right) - \frac{3m}{m - s + 1} \frac{r^s}{mN}.
\]

(21)

The proof of the above theorem follows by just combining the ingredients we have developed so far and is similar to that of Theorem 5.1 with the use of Observation 6.6. The details can be found in the full version.

Finally, all that is left to be done is to pick parameters to show how the above can lead to optimal rate list-decodable codes over a constant-sized alphabet which further achieve very good listsizes.

Let \( s > 0 \) be a small positive constant, and a family of codes of length \( N \) (assumed large enough) and rate \( R \in (0, 1) \) is sought. Pick \( n \) to be a growing parameter.

Let us pick \( s = \Theta(1/\varepsilon) \), \( m = \Theta(1/\varepsilon^2) \), \( \zeta = s/\varepsilon \), \( r = \Theta(1/\varepsilon) \), \( q = r^2 \), and \( e = \left\lceil \frac{\log q}{\log r} \right\rceil \), \( N = \left\lceil \frac{r}{(r-1)^2} \right\rceil \), and \( k = R N(1 + \varepsilon) \). This ensures that (i) there are at least \( Nm \) rational places and so we get a code of length at least \( n/m = N \), (ii) the rate of the code \( C_2 \) is at least \( R \), and (iii) the error fraction (21) is at least \( 1 - R - \varepsilon \).

The remaining part is to pick a multiple \( \Delta \) of \((r-1) \) so that the \( k < q^{\frac{\Delta}{2}} \) condition is met. This can be achieved by choosing \( u = \left\lceil \frac{\log q/\log r}{\log \Delta} \right\rceil \) and \( \Delta = (r-1)u \). With these choices, we can conclude the following, which is the main final result of this paper.

Theorem 6.8. (Main; Corollary to Theorem 6.7 with above choice of parameters) For any \( R \in (0, 1) \) and positive constant \( \varepsilon \in (0, 1) \), there is a Monte Carlo construction of a family of codes of rate at least \( R \) over an alphabet...
size $\exp(O((1/\varepsilon)/\varepsilon^2))$ that are encodable and $(1 - R - \varepsilon, O(1/(R\varepsilon)))$-list decodable in poly($N$) time, where $N$ is the block length of the code.

It may be instructive to recap why the Hermitian tower could not give a result like the above one. In the Hermitian case, the ratio $\varphi/\nu$ of the genus to the number of rational places was about $e/r = e/\sqrt{q}$, and thus we needed $q > e^2$. Since the period $\Delta$ was about $q$, the running time of the decoder was bigger than $q^{O(q)}$, whereas the length of the code was at most $q^{O(\sqrt{q})}$. This dictated the choice of $q \approx \log^2 n$, and then to keep the running time polynomial, we had to take $\zeta \approx (\log n \log \log n)^{-1}$.

7. REFERENCES

[1] Z. Dvir and S. Lovett. Subspace-evasive sets. *Electronic Colloquium on Computational Complexity*, TR11-139, 2011.

[2] P. Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37:5–12, 1991.

[3] A. Garcia and H. Stichtenoth. A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vladut bound. *Inventiones Mathematicae*, 121:211–222, 1995.

[4] A. Garcia and H. Stichtenoth. On the asymptotic behavior of some towers of function fields over finite fields. *Journal of Number Theory*, 61(2):248–273, 1996.

[5] V. Guruswami. Cyclotomic function fields, Artin-Frobenius automorphisms, and list error-correction with optimal rate. *Algebra and Number Theory*, 4(4):433–463, 2010.

[6] V. Guruswami. Linear-algebraic list decoding of folded Reed-Solomon codes. In *Proceedings of the 26th IEEE Conference on Computational Complexity*, June 2011.

[7] V. Guruswami and S. Narayanan. Combinatorial limitations of a strong form of list decoding. *CoRR*, abs/1202.6086, 2012.

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

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

[10] V. Guruswami and C. Wang. Optimal rate list decoding via derivative codes. In *Proceedings of APPROX/RANDOM 2011*, pages 593–604, August 2011.

[11] F. Parvaresh and A. Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 285–294, 2005.

[12] B. Z. Shen. A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate. *IEEE Transactions on Information Theory*, 39:239–242, 1993.

[13] K. Shum, I. Aleshnikov, P. V. Kumar, H. Stichtenoth, and V. Deolalikar. A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound. *IEEE Transactions on Information Theory*, 47(6):2225–2241, 2001.

[14] H. Stichtenoth. *Algebraic Function Fields and Codes*. Universitext, Springer-Verlag, Berlin, 1993.

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

[16] S. Vadhan. *Pseudorandomness*. Foundations and Trends in Theoretical Computer Science (FnT-TCS). NOW publishers, 2010. To appear. Draft available at http://people.seas.harvard.edu/~salil/pseudorandomness/.