NEARLY LINEAR TIME ENCODABLE CODES BEATING THE GILBERT-VARSHAMOV BOUND

ANAND KUMAR NARAYANAN[1] AND MATTHEW WEIDNER[2]

Abstract. We construct explicit nearly linear time encodable error-correcting codes beating the Gilbert-Varshamov bound. Our codes are algebraic geometry codes built from the Garcia-Stichtenoth function field tower and beat the Gilbert-Varshamov bound for alphabet sizes at least 19^2. Messages are identified with functions in certain Riemann-Roch spaces associated with divisors supported on multiple places. Encoding amounts to evaluating these functions at degree one places. We devise an intricate deterministic nearly linear time encoding algorithm by exploiting algebraic structures particular to the Garcia-Stichtenoth tower. Further, we present fast unique and list decoding algorithms with runtime exponents equal to the matrix multiplication exponent $\omega < 2.373$.

[1] Laboratoire d'Informatique de Paris 6, Pierre et Marie Curie University, Paris.
[2] Department of Mathematics, California Institute of Technology, Pasadena.
E-mail addresses: anand.narayanan@lip6.fr, mweidner@caltech.edu.
Supported by NSF grant #CCF-1423544 and Chris Umans' Simons Foundation Investigator grant.

1. Introduction

1.1. **Codes Beating the Gilbert-Varshamov Bound.** Error-correcting codes enable reliable transmission of information over an erroneous channel. A (block) error-correcting code of block length N over a finite alphabet Σ of size Q is a subset $\mathcal{C} \subseteq \Sigma^N$. The rate R at which information is transmitted through the code $\mathcal{C}$ is defined as $\log_Q(|\mathcal{C}|)/N$. The minimum distance d of the code $\mathcal{C}$, defined as the minimum Hamming distance among all distinct pairs of elements (codewords) in the code $\mathcal{C}$, quantifies the number of errors that can be tolerated. A code with minimum distance d can tolerate $(d-1)/2$ errors. The relative distance δ is defined as $\delta := d/N$. A code $\mathcal{C}$ is linear if the alphabet is a finite field $\mathbb{F}_Q$ (with Q elements) and $\mathcal{C}$ is an $\mathbb{F}_Q$-linear subspace of $\mathbb{F}_Q^N$.

One typically desires codes to transmit information at a high rate while still being able to correct a large fraction of errors. That is, one wants codes with large rate and large relative distance. However, rate and relative distance are competing quantities with a tradeoff between them. The Gilbert-Varshamov bound assures, for every Q, $r > 0$, $0 < \delta \leq 1 - 1/Q$ and small positive ϵ, the existence of an infinite family of codes with increasing block length over an alphabet of size Q with rate r and relative distance δ bounded by

$$R + H_Q(\delta) \geq 1 - \epsilon, \tag{1}$$

where H_Q is the Q-ary entropy function [6, 20]. Random codes, where one chooses a random subset of N-tuples over the alphabet as the code, achieve the Gilbert-Varshamov bound with equality, with high probability. Likewise, random linear codes (where one chooses a random subspace of N-tuples over a finite field) meet the bound with high probability. In fact, Varshamov proved the bound using the probabilistic method with random linear codes. Testing if a given linear code meets the Gilbert-Varshamov bound comes down to approximating the minimum distance, a task that is NP-hard, even on average. Hence constructing codes meeting or beating the Gilbert-Varshamov bound remained a long-standing open problem, until the advent of algebraic geometry codes.

Goppa proposed algebraic geometry codes obtained from curves over finite fields as a generalization of Reed-Solomon codes [7]. Messages are identified with functions on the curve in the Riemann-Roch space corresponding to a chosen divisor with support disjoint from a large set of $\mathbb{F}_Q$-rational points on the curve. Evaluations of functions in the Riemann-Roch space at these $\mathbb{F}_Q$-rational points on the curve is taken as the code. The rate of the code is the ratio of the dimension of the Riemann-Roch space to the number of $\mathbb{F}_Q$-rational points. The Riemann-Roch theorem gives a bound on the dimension of the code and yields the following tradeoff between rate and relative distance:

$$R + \delta \geq 1 - \frac{g}{N}, \tag{2}$$

where g denotes the genus of the curve. This spurred an effort to construct curves over finite fields where the fraction g/N of the genus to the number of $\mathbb{F}_Q$-rational points is as low as possible. Researchers had to contend with the lower bound $g/N \geq 1/(\sqrt{Q} - 1)$ of Drinfeld-Vlăduţ [3]. In a seminal paper, Tsfasman, Vlăduţ and Zink constructed curves meeting the Drinfeld-Vlăduţ bound when the underlying finite field size Q is a square, leading to the bound [19]

$$R + \delta \geq 1 - \frac{1}{\sqrt{Q} - 1}. \tag{3}$$

Remarkably, for $Q \geq 7^2$, the Tsfasman-Vlăduţ-Zink bound is better than the Gilbert-Varshamov bound! This is a rare occasion where an explicit construction yields better parameters than guaranteed by randomized arguments. Garcia and Stichtenoth described an explicit tower of function fields which meet the Drinfeld-Vlăduţ bound, hence yield codes matching the Tsfasman-Vlăduţ-Zink bound [5]. The curves in the Garcia-Stichtenoth tower are the primary objects of study in our paper. An outstanding open problem in this area is to explicitly construct codes meeting or beating the Gilbert-Varshamov bound over small alphabets, in particular binary codes ($Q = 2$).

1.2. **Linear Time Encodable Codes meeting the Gilbert-Varshamov Bound.** For codes to find use in practice, one often requires fast encoding and decoding algorithms in addition to satisfying a good tradeoff between rate and minimum distance. An encoding algorithm maps a given message to a codeword. A decoding algorithm takes a possibly corrupted codeword, called the received word, and outputs the message that induced it, provided the number of errors is within the designed tolerance.

A natural question, which remains unresolved, is if there exist linear time encodable and decodable codes meeting or beating the Gilbert-Varshamov bound. One cannot look to random linear codes to resolve this problem, for they require quadratic runtime to encode and are NP-hard to decode. In a breakthrough, Speilman, using explicit expander codes, proved the existence of linear time encodable and decodable "good" codes [17]. A family of codes with increasing block length is deemed good if (in the limit) the rate and relative distance are simultaneously bounded away from zero. A code being good is a weaker condition than meeting the Gilbert-Varshamov bound. Guruswami and Indyk constructed linear time encodable and decodable expander codes approaching the Gilbert-Varshamov bound [10]. However, the closer one wishes to approach the Gilbert-Varshamov bound, the larger the alphabet size of the code. Druk and Ishai constructed linear time encodable codes meeting the Gilbert-Varshamov bound, but these codes are likely NP-hard to decode [4].

1.3. **Nearly Linear Time Encodable Codes Beating the Gilbert-Varshamov Bound.** Our main result is the explicit algebraic construction of nearly linear time encodable codes beating the Gilbert-Varshamov bound, along with an efficient decoding algorithm.

Theorem 1.1. *For every square prime power Q and rate $R \in (0, 1 - \frac{2\sqrt{Q}+1}{Q-\sqrt{Q}})$, there exists an infinite sequence of codes over $\mathbb{F}_Q$ of increasing length N with rate R and relative distance δ satisfying*

$$R + \delta \geq 1 - \frac{2\sqrt{Q}+1}{Q-\sqrt{Q}}.$$

Further, there exists deterministic algorithms to:

- *pre-compute a representation of the code at the encoder and decoder in $O(N^{3/2}\log^3 N)$ time; this representation occupies $O(N)$ space.*
- *encode a message in $O(N^{\omega/2})$ time, where ω is the matrix multiplication exponent.*
- *decode close to (and list decode beyond) half the relative designed distance in $O(N^{\omega})$ time.*

For $Q \geq 19^2$, the tradeoff assured by Theorem 1.1 is better than the Gilbert-Varshamov bound. The encoding time is linear if the matrix multiplication exponent ω is indeed 2 as widely conjectured. The best known bound for ω yields an encoding time exponent of 1.19 [13].

The pre-processing step can be thought of as computing a succinct representation of the code and is performed at the encoder and decoder independently. If one desires, the pre-processing and encoding time can be made to approach linear time at the cost of needing larger alphabets to beat the Gilbert-Varshamov bound. Our construction is parametrized by an integer $k \geq 2$ and yields the tradeoff

$$R + \delta \geq 1 - \frac{k\sqrt{Q}+k-1}{Q-\sqrt{Q}}$$

with pre-computation requiring time $O(N^{3/k})$, the resulting succinct representation requiring $O(N^{2/k})$ space, and encoding requiring time $O(N^{1+(\omega-2)/k})$. Theorem 1.1 corresponds to $k = 2$. See Table 1 for a comparison of $k = 2, 3$. The likeness to the Tsfasman-Vlăduţ-Zink bound (equation 3), which is obtained if one is allowed to substitute $k = 1$, is striking.

We next recount algebraic geometry codes before sketching our construction. Let $\mathcal{X}$ be a (not necessarily plane) curve over a finite field $\mathbb{F}_Q$ and $\mathbb{F}_Q(\mathcal{X})$ the associated function field. A set $\mathcal{P}$ of $\mathbb{F}_Q$-rational points on $\mathcal{X}$ (or equivalently, degree 1 places in $\mathbb{F}_Q(\mathcal{X})$) will serve as the code places. A divisor is chosen, typically of the form rP_∞ where P_∞ is a place in $\mathbb{F}_Q(\mathcal{X})$ away from $\mathcal{P}$ and $r \in \mathbb{N}$. The Riemann-Roch space $\mathcal{L}(rP_\infty)$ (consisting of functions in $\mathbb{F}_Q(\mathcal{X})$ whose poles are confined to P_∞ and have order bounded by r) is identified with the message space. The code is the evaluation of $\mathcal{L}(rP_\infty)$ at the places in $\mathcal{P}$. The rate is determined by the dimension of $\mathcal{L}(rP_\infty)$ as an $\mathbb{F}_Q$-linear space. The Riemann-Roch theorem then yields bounds on the rate and relative distance, thereby quantifying the performance of the code. Encoding messages requires efficient algorithms to construct and evaluate functions from the Riemann-Roch space. This can be accomplished in polynomial time due to algorithms of Huang and Ierardi (for smooth projective plane curves) [12] and Hess (for smooth projective curves) [11, 9]. However, such generic algorithms are far from linear. We focus on building fast algorithms tailored to the Garcia-Stichtenoth tower.

Take $Q = q^2$ where q is a prime power. The Garcia-Stichtenoth tower over $\mathbb{F}_{q^2}$ is the sequence of function fields defined by $F_0 = \mathbb{F}_{q^2}(x_0)$, and $F_{i+1} = F_i(x_{i+1})$ where x_{i+1} satisfies the relation

$$x_{i+1}^q + x_{i+1} = \frac{x_i^q}{x_i^{q-1} + 1}.$$

Let $P_\infty^{(n)}$ denote the unique pole of x_0 in F_n. In a series of works, Aleshnikov, Deolalikar, Kumar, Shum and Stichtenoth [1, 15, 16] described the splitting of places in F_n and established a pole-cancelling algorithm to compute a basis for the Riemann-Roch spaces $\mathcal{L}(rP_\infty^{(n)})$. This culminated in a quadratic time algorithm (with nearly cubic time pre-processing) to encode these codes.

From the n^{th} function field F_n, we construct codes of block length $q^n(q^2 - q)$. Let a small integer parameter $k \geq 2$ be chosen. Assume for ease of exposition that k divides n.

Code places: Let $\Omega = \{\alpha \in \mathbb{F}_{q^2} \mid \alpha^q + \alpha = 0\}$, which has size q. The code places are all the places in F_n which are zeros of $x_0 - \alpha$ for $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$; there are $q^n(q^2 - q)$ such places, all of which are $\mathbb{F}_{q^2}$-rational.

Message Space: We begin by constructing functions in the lower function field $F_{n/k}$ that are regular. By regular, we mean that their poles in F_n are confined to $P_\infty^{(n)}$. We devise a procedure called *shifting* that translates functions in $F_{n/k}$ to F_n. It is quite simple and, in spirit, just relabelling the subscripts so that each x_j becomes x_{j+i} for some chosen positive integer i. The symmetry of the defining equations of the Garcia-Stichtenoth tower allows us to determine the pole divisor in F_n of shifts of regular functions from $F_{n/k}$. The poles of the shifts are not confined to $P_\infty^{(n)}$. However, by taking products of carefully chosen shifted functions, we can bound the new poles arising outside $P_\infty^{(n)}$. We thus construct mostly regular functions in F_n by taking products of shifts of regular functions in $F_{n/k}$.

In summary, given a positive integer r, we can construct a large number of functions in a Riemann-Roch space of the form $\mathcal{L}(G + rP_\infty^{(n)})$ where G is a small pole divisor. We take the span of these constructed functions to be our message space. Enough functions are constructed to yield large rate codes, and properties of curves applied to the divisor $G + rP_\infty^{(n)}$ yield a lower bound on the minimum distance.

Pre-computation: To aid in rapid encoding and decoding, we first pre-compute a set of regular functions in $F_{n/k}$. In particular, we pre-compute the evaluations at all code places of a basis for all regular functions in $F_{n/k}$ of a certain bounded pole degree. This Riemann-Roch space computation is performed using an algorithm of Shum et. al. [15] and has runtime exponent $3/k$.

Encoding Algorithm: Given such a set of regular functions in $F_{n/k}$, we compute basis functions in our message space by taking products of shifted functions. Given a message, which is a tuple over $\mathbb{F}_{q^2}$, encoding amounts to evaluating the corresponding linear combination of the basis functions simultaneously at the code places. We devise a Baby-Step Giant-Step algorithm to perform this multipoint evaluation. The runtime of the encoding step depends on the parameter k. For $k = 2$, the crux of the computation is square matrix multiplications, resulting in an encoding algorithm with runtime exponent $\omega/2$. For larger values of k, the crux is rectangular matrix multiplications of shape determined by k, and the runtime is again nearly linear. In particular, larger values of k give rise to faster encoding algorithms.

Decoding Algorithms: We tailor the Shokrollahi-Wasserman algorithm [14] to our code setting to uniquely decode close to half the relative distance and list decode beyond that. The Shokrollahi-Wasserman algorithm first interpolates a polynomial with coefficients in a Riemann-Roch space such that each message sufficiently close to the received word is a root. Then the roots of the interpolated polynomial in the message Riemann-Roch space are enumerated. Finally the encoding algorithm is used to verify and output the messages in the enumeration that are indeed sufficiently close to the received word. To adapt their algorithm to our setting, we identify an appropriate Riemann-Roch space (which incidentally is an extension of the message space) as the coefficient space of the interpolation polynomial. To perform root finding, we deploy the Hensel lifting algorithm of Augot-Pecquet [2]. An algorithmic ingredient required in both the interpolation and root finding steps is a means to evaluate basis functions of the coefficient and message spaces at the evaluation places. Our encoding algorithm efficiently provides this ingredient, so the bottleneck in our algorithm is the generic linear algebra in the interpolation step, leading to a runtime exponent of ω.

1.4. **Organization:** In §2, we recount results from [1] on the splitting of places in the Garcia-Stichtenoth tower and establish notation. In §3, we define the shifting operation and construct mostly regular functions in F_n from regular functions in $F_{n/k}$. The code sequences derived from mostly regular functions are defined and their parameters established in §4. In §5 we develop the nearly linear time encoding algorithm using fast matrix multiplication. For want of space, we move the decoding algorithm to the appendix (§7).

2. Splitting of Places in the Garcia-Stichtenoth Tower

In this section, we recall some notation and results from [1] on the splitting of places in the Garcia-Stichtenoth tower.

In $F_0 = \mathbb{F}_{q^2}(x_0)$, x_0 has a unique pole, which we denote by $P_\infty^{(0)}$. This place is totally ramified in every field extension F_n/F_0, hence there is a unique place lying above $P_\infty^{(0)}$ in F_n; we denote this place by $P_\infty^{(n)}$. Let $\Omega = \{\alpha \in \mathbb{F}_{q^2} \mid \alpha^q + \alpha = 0\}$, which has size q. For $\alpha \in \mathbb{F}_{q^2}$, let $P_\alpha^{(0)}$ denote the unique zero of $x_0 - \alpha$ in F_0. When $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$, $P_\alpha^{(0)}$ splits completely in every field extension F_n/F_0, yielding q^n $\mathbb{F}_{q^2}$-rational places. As α varies, we get $(q^2 - q)q^n$ rational places in F_n, which we take to be the set of code places.

When $\alpha \in \Omega \setminus \{0\}$, $P_\alpha^{(0)}$ is totally ramified in every field extension F_n/F_0, hence there is a unique place lying above it in F_n; we denote this place by $P_\alpha^{(n)}$. The most interesting place is $P_0^{(0)}$. For $t \geq 1$, let $S_t^{(t-1)}$ denote the unique place in F_{t-1} that is a zero of x_{t-1}. We sometimes treat $S_t^{(t-1)}$

as a singleton set instead of a place. We have $S_1^{(0)} = P_0^{(0)}$, and $S_u^{(u-1)}$ lies over $S_t^{(t-1)}$ whenever $u \geq t$.

In the field extension F_t/F_{t-1}, $S_t^{(t-1)}$ splits completely. Specifically, for each $\alpha \in \Omega$, there is a unique place of F_t which is a simultaneous zero of x_{t-1} and $x_t - \alpha$, and these are all of the places lying above $S_t^{(t-1)}$. We let $S_t^{(t)}$ denote the set of all places lying above $S_t^{(t-1)}$ in F_t besides $S_{t+1}^{(t)}$, i.e., $S_t^{(t)}$ contains the simultaneous zero of x_{t-1} and $x_t - \alpha$ for each $\alpha \in \Omega \setminus \{0\}$. For $u \geq t$, we let $S_t^{(u)}$ denote the set of all places of F_u lying above a place in $S_t^{(t)}$. For $t \leq t' \leq u+1$, we let $S_{t,t'}^{(u)} := \bigcup_{i=t}^{t'} S_i^{(u)}$.

Let $u \geq t-1$. If $u < 2t$, then all of the places in $S_t^{(u)}$ are unramified (but not necessarily split) in the field extension F_{u+1}/F_u. If $u \geq 2t$, then all of the places in $S_t^{(u)}$ are totally ramified in F_{u+1}/F_u.

If Q is a place of F_n and $x \in F_n$, we let $v_Q(x)$ denote the valuation of x at Q. We define the weight of $x \in F_n$ to be $-v_{P_\infty^{(n)}}(x)$.

3. Mostly Regular Functions Through Shifting

For the remainder of the paper, fix an integer parameter $k \geq 2$. In this section, given a positive integer r, through shifting we construct a mostly regular function $f_r \in F_n$ of weight precisely $r + \sum_{i=1}^{k} q^{n-(i-1)\lceil n/k \rceil + 1}$ from regular functions in $F_{n/k}$. The discrepancy of f_r from being regular will be quantified by a pole divisor G of degree at most $q^n(kq + k - 1)$. That is, there is a pole divisor G of said degree and weight $\sum_{i=1}^{k} q^{n-(i-1)\lceil n/k \rceil + 1}$ such that for all r,

$$f_r \in \mathcal{L}(G + r(P_\infty^{(n)})) \setminus \mathcal{L}(G + (r-1)(P_\infty^{(n)})).$$

Once a choice of a regular function of each weight in $F_{n/k}$ used by our construction is fixed, the functions f_r are uniquely determined.

3.1. Shifting.

First, we define the shifting operation and determine the poles in $\mathbb{F}_n$ of functions arising out of shifting regular functions in $F_{n/k}$.

Let $f = x_m^{e_m} \cdots x_0^{e_0}$ be a monomial in F_m. For $i \geq 0$, we define the *shift of f by i* to be the element

$$f[i] := x_{m+i}^{e_m} \cdots x_i^{e_0} \in F_{m+i}.$$

We extend the definition of shift $\mathbb{F}_{q^2}$-linearly to all of F_m.

For $x \in F_m$ and $n \geq m$, we use $(x)^{(n)}$ and $(x)_\infty^{(n)}$ to denote the principal divisor and pole divisor of x as an element of F_n.

Proposition 3.1. *(a) For any $f \in F_m$ and $i \geq 0$, $f[i]$ is well-defined, i.e., it does not depend on the representation of f as a sum of monomials.*

(b) Let $f \in F_m$ be regular of weight r. Then:

- *$f[i]$ has weight r.*
- *$f[i]$ is regular at $S_{i,m+i+1}^{(m+i)}$.*
- *For $t \in [0, i-1]$, for all $P \in S_t^{(m+i)}$, we have*

$$v_P(f[i]) = \begin{cases} -r & \text{if } t \leq \frac{m+i}{2} \\ -rq^{2t-(m+i)} & \text{if } t > \frac{m+i}{2}. \end{cases}$$

- $\deg\left((f[-i])_\infty^{(m+i)}\right) = rq^i$.

Proof. For all j, we have the isomorphism

$$\begin{aligned}\phi_j : F_j &\xrightarrow{\sim} F_j \\ x_k &\mapsto x_{j-k}^{-1},\end{aligned}$$

which is its own inverse [15, p. 2237]. It is easy to see that

$$f[i] = \phi_{m+i}(\phi_i(f)),$$

proving (a).

To prove (b), we use the fact that ϕ_j induces bijections $S_t^{(j)} \leftrightarrow S_{j-t}^{(j)}$ for each $t \in [0, j]$, together with a correspondence $P_\infty^{(j)} \leftrightarrow S_{j+1}^{(j)}$ [15, p. 2237]. Thus letting $f \in F_m$ be regular of weight r, $\phi_m(f)$ is regular at all places (including $P_\infty^{(j)}$) except for a pole of order r at $S_{j+1}^{(j)}$. Then by the ramification behavior of the tower, $\phi_m(f) \in F_{m+i}$ is regular at all places (including $P_\infty^{(j)}$) except for $S_{m,m+i+1}^{(m+i)}$. In particular, for $t \in [m, m+i+1]$, for all $P \in S_t^{(m+i)}$, we have

$$v_P(\phi_m(f)) = \begin{cases} -r & \text{if } t \geq \frac{m+i}{2} \\ -rq^{m+i-2t} & \text{if } t < \frac{m+i}{2} \end{cases}$$

Then $f[i] = \phi_{m+i}(\phi_m(f))$ easily has the first three properties in (b). The fourth property follows either from computing the total pole degree directly, or from noting that

$$\begin{aligned}\deg\left((f[i])_\infty^{(m+i)}\right) &= \deg\left((\phi_m(f))_\infty^{(m+i)}\right) \\ &= [F_{m+i} : F_m]\deg\left((\phi_m(f))_\infty^{(m)}\right) \\ &= q^i \deg\left((f)_\infty^{(m)}\right) \\ &= rq^i.\end{aligned}$$

□

3.2. **Construction of Mostly Regular Functions.** We begin by dealing with the case $r \in [0, q^n - 1]$. Write $r = r_1 q^{n-\lceil n/k\rceil} + r_2 q^{n-2\lceil n/k\rceil} + \cdots + r_{k-1} q^{n-(k-1)\lceil n/k\rceil} + r_k$, with $r_1, \ldots, r_{k-1} \in [0, q^{\lceil n/k\rceil} - 1]$ and $r_k \in [0, q^{n-(k-1)\lceil n/k\rceil} - 1]$. For $i \in [1, k-1]$, let $\bar{f}_i \in F_{\lceil n/k\rceil}$ be regular of weight $q^{\lceil n/k\rceil + 1} + r_i$, and let $\bar{f}_k \in F_{n-(k-1)\lceil n/k\rceil}$ be regular of weight $q^{n-(k-1)\lceil n/k\rceil+1} + r_k$. Set

$$f_r := \prod_{i=1}^{k} \bar{f}_i[(i-1)\lceil n/k\rceil].$$

The following proposition shows that f_r is mostly regular with weight precisely $r + \sum_{i=1}^k q^{n-(i-1)\lceil n/k\rceil+1}$.

Proposition 3.2. *There exists a pole divisor G of degree at most $q^n(kq + k - 1)$ and weight $\sum_{i=1}^k q^{n-(i-1)\lceil n/k\rceil+1}$ such that for all $r \in [0, q^n - 1]$,*

$$f_r \in \mathcal{L}(G + r(P_\infty^{(n)})) \setminus \mathcal{L}(G + (r-1)(P_\infty^{(n)})).$$

Proof. Using proposition 3.1, it is easy to see that f_r has weight $r + \sum_{i=1}^k q^{n-(i-1)\lceil n/k\rceil+1}$ and is regular outside of $S_{0,(k-1)\lceil n/k\rceil-1}^{(n)}$. It remains to bound the pole orders at the places in $S_{0,(k-1)\lceil n/k\rceil-1}^{(n)}$.

Let $\bar{f}_1, \ldots, \bar{f}_k$ be as in the definition of f_r. For $i \in [1, k-1]$, $\bar{f}_i$ has weight less than $q^{\lceil n/k \rceil+1} + q^{\lceil n/k \rceil}$ in $F_{\lceil n/k \rceil}$. Thus by the above proposition, the pole divisor of $\bar{f}_i[(i-1)\lceil n/k \rceil]$ in $F_{i\lceil n/k \rceil}$ satisfies

$$(\bar{f}_i[(i-1)\lceil n/k\rceil])_\infty^{(i\lceil n/k\rceil)} - (q^{\lceil n/k\rceil+1}+r_i)P_\infty^{(i\lceil n/k\rceil)} \le (q+1)q^{\lceil n/k\rceil} \sum_{t=0}^{(i-1)\lceil n/k\rceil-1} q^{\max\{0,2t-i\lceil n/k\rceil\}} S_t^{(i\lceil n/k\rceil)}.$$

Here we use $S_t^{(i\lceil n/k\rceil)}$ as a shorthand for the divisor $\sum_{P \in S_t^{(i\lceil n/k\rceil)}} P$. Let G_i denote the divisor on the right-hand side. By direct computation or by the same trick used in the proof of Proposition 3.1, we have $\deg(G_i) = (q+1)(q^{i\lceil n/k\rceil} - q^{\lceil n/k\rceil})$.

Next, $\bar{f}_k$ has weight less than $q^{n-(k-1)\lceil n/k\rceil+1} + q^{n-(k-1)\lceil n/k\rceil}$ in $F_{n-(k-1)\lceil n/k\rceil}$. Hence again

$$(\bar{f}_k[(k-1)\lceil n/k\rceil])_\infty^{(n)} - (q^{n-(k-1)\lceil n/k\rceil+1}+r_k)P_\infty^{(n)} \le (q+1)q^{n-(k-1)\lceil n/k\rceil} \sum_{t=0}^{(k-1)\lceil n/k\rceil-1} q^{\max\{0,2t-n\}} S_t^{(i\lceil n/k\rceil)}.$$

Let G_k denote the divisor on the right-hand side. As above, we have $\deg(G_k) = (q+1)(q^n - q^{n-(k-1)\lceil n/k\rceil})$.

For a finite extension of function fields E'/E, we let $\mathrm{Con}_E^{E'}$ denote the corresponding conorm map; this is the unique homomorphism from the divisor group of E to the divisor group of E' such that for all places Q of E,

$$\mathrm{Con}_E^{E'}(Q) = \sum_{Q'|Q} e(Q'|Q) \cdot Q',$$

where the sum runs over all places Q' of $F_{E'}$ lying over Q and where $e(Q'|Q)$ denotes the ramification index of Q' over Q. We have the identities

$$(x)^{E'} = \mathrm{Con}_E^{E'}\left((x)^E\right)$$
$$(x)_\infty^{E'} = \mathrm{Con}_E^{E'}\left((x)_\infty^E\right)$$

for all $x \in E$, where $(x)^E$ and $(x)_\infty^E$ denote the divisor and pole divisor of x as an element of E, and similarly for E'. Also, for all divisors D of E, we have $\deg \mathrm{Con}_E^{E'}(D) = [E' : E] \deg(D)$.

Then using the pole divisors computed above, it is easy to see that

$$(f_r)_\infty^{(n)} - v_{P_\infty^{(n)}}(f_r) P_\infty^{(n)} \le \sum_{i=1}^{k-1} \mathrm{Con}_{F_{i\lceil n/k\rceil}}^{F_n} G_i + G_k.$$

Define

$$G = \left(\sum_{i=1}^{k} q^{n-(i-1)\lceil n/k\rceil+1}\right) P_\infty^{(n)} + \sum_{i=1}^{k-1} \mathrm{Con}_{F_{i\lceil n/k\rceil}}^{F_n} G_i + G_k.$$

Then the above remarks show that

$$f_r \in \mathcal{L}(G + r(P_\infty^{(n)})) \setminus \mathcal{L}(G + (r-1)(P_\infty^{(n)})),$$

and

$$\begin{aligned}\deg(G) &= \sum_{i=1}^{k} q^{n-(i-1)\lceil n/k\rceil+1} + \sum_{i=1}^{k-1}(q+1)(q^n - q^{n-(i-1)\lceil n/k\rceil}) + (q+1)(q^n - q^{n-(k-1)\lceil n/k\rceil}) \\ &\le k(q+1)q^n - q^n = q^n(kq+k-1).\end{aligned}$$

□

For general $r \geq 0$, say $r = sq^n + t$ with $t \in [0, q^n - 1]$, set

$$f_r := x_0^s f_t.$$

Then we again have $f_r \in \mathcal{L}(G + r(P_\infty^{(n)})) \setminus \mathcal{L}(G + (r-1)(P_\infty^{(n)}))$ because x_0 is regular of weight q^n.

4. Code Sequences Beating the Gilbert-Varshamov Bound

Define an $\mathbb{F}_{q^2}$-linear map $\psi : \mathbb{F}_{q^2}^{\mathbb{N}^0} \to \bigcup_r \mathcal{L}(G + rP_\infty^{(n)})$ by sending the r-th basis vector to f_r (we zero-index the basis vectors). Because the f_r have distinct weights, the strict triangle inequality implies that ψ is injective.

Each f_r is regular at all of the code places. Hence we can speak of the evaluation map ev : $\bigcup_r \mathcal{L}(G + rP_\infty^{(n)}) \to \mathbb{F}_{q^2}^{q^n(q^2-q)}$, which maps a function to the tuple of its values at the code places.

Proposition 4.1. *Let $K \in [1, q^n(q^2 - q - kq - k + 1)]$. Then the map $(\mathrm{ev} \circ \psi)|_{\mathbb{F}_{q^2}^K} : \mathbb{F}_{q^2}^K \to \mathbb{F}_{q^2}^{q^n(q^2-q)}$ is injective, and its image defines an $[N, K, D]$ code, where $N = q^n(q^2 - q)$ and*

$$D \geq N - K - q^n(kq + k - 1) + 1.$$

Proof. This follows from the above bound on $\deg(G)$ and a standard argument about algebraic geometry codes. For completeness, we give the proof in full.

Let $D^* := N - K - q^n(kq + k - 1) + 1$. Suppose that for some nonzero $v \in \mathbb{F}_{q^2}^K$, the N-tuple $\mathrm{ev}(\psi(v))$ has less than D^* nonzero coordinates. Let $M > N - D^*$ be the number of coordinates which are zero. We already know that $\psi(v) \in \mathcal{L}(G + (K-1)P_\infty^{(n)})$. By definition of M, there are code places $P_1, \ldots, P_M$ at which $\psi(v)$ is zero. Then $\psi(v)$ lies in the Riemann-Roch space $\mathcal{L}(D)$, where

$$D = G + (K-1)P_\infty^{(n)} - \sum_{i=1}^{M} P_i.$$

But $\deg(D) = \deg(G) + K - 1 - M < q^n(kq + k - 1) + K - 1 - N + D^* = 0$ by Proposition 3.2 and the definition of D^*, so $\mathcal{L}(D) = \{0\}$ and $\psi(v) = 0$. But we said above that ψ is injective, so this is a contradiction.

To see that $(\mathrm{ev} \circ \psi)|_{\mathbb{F}_{q^2}^K}$ is injective, note that for $K \leq q^n(q^2 - q - kq - k + 1)$, we have $D^* \geq 1$, hence the above argument shows that at least one coordinate of $\mathrm{ev}(\psi(v))$ is nonzero. □

The above proposition implies that for any $n \geq 0$, we can define codes of the above form with length $q^n(q^2 - q)$ over $\mathbb{F}_{q^2}$ whose rate R and relative distance δ satisfy

$$R + \delta \geq 1 - \frac{kq + k - 1}{q^2 - q},$$

with many choices of rate. For all k, this exceeds the Gilbert-Varshamov bound for large enough q.

5. Nearly Linear Time Encoding

The encoding task is: given a message $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$, output $\mathrm{ev}(\psi(v))$. For simplicity, we assume throughout this section that k divides n; this affects the runtime by a factor of at most $\mathrm{poly}(q)$, which is a constant in our context. Our approach is to first write the encoding of a vector $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$ with respect to $F_{i(n/k)}$ in terms of some encodings with respect to $F_{(i-1)(n/k)}$ and

$F_{n/k}$. We then use a Baby-Step Giant-Step algorithm and fast matrix multiplication to build up the encoding of v starting from encodings with respect to $F_{n/k}$.

5.1. **Pre-computation.** Pre-compute the evaluations of some $g_0, \ldots, g_{q^{n/k}-1} \in F_{n/k}$ at the code places of $F_{n/k}$, where each g_s is regular of weight $q^{n/k+1}+s$. This can be done using the deterministic algorithm in [15].

5.2. **Nearly Linear Time Encoding with Fast Matrix Multiplication.** We begin by considering encoding for $v \in \mathbb{F}_{q^2}^{q^n}$. Encoding messages of length greater than q^n will be dealt with at the end of this subsection.

For $i \in [0, k]$, $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$, and P a code place of $F_{i(n/k)}$, we set $w(P) := \psi_i(w)(P)$, where ψ_i is the function ψ corresponding to $F_{i(n/k)}$.

Proposition 5.1. *Let $i \in [1, k]$, let $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$, and let P be a code place of $F_{i(n/k)}$. Uniquely write*

$$w = \sum_{l=0}^{q^{n/k}-1} \iota_l(w^{(l)})$$

for $w^{(l)} \in \mathbb{F}_{q^2}^{q^{(i-1)(n/k)}}$, where $\iota_l : \mathbb{F}_{q^2}^{q^{(i-1)(n/k)}} \hookrightarrow \mathbb{F}_{q^2}^{q^{i(n/k)}}$ is the vector space embedding sending the j-th basis vector to the $(j + lq^{(i-1)(n/k)})$-th basis vector. Let P' denote the place obtained by restricting P to $F_{(i-1)(n/k)}$, and let P'' denote the place of $F_{n/k}$ at which x_0 has value $x_{(i-1)(n/k)}(P)$, x_1 has value $x_{(i-1)(n/k)+1}(P)$, etc. Then P' and P'' are code places, and

$$w(P) = \sum_{l=0}^{q^{n/k}-1} w^{(l)}(P')g_l(P'').$$

Proof. By [5, Lemma 3.9], the code places of F_m are precisely the places at which each x_t has value in $\mathbb{F}_{q^2} \setminus \Omega$, subject to the relations defining the Garcia-Stichtenoth tower. Both P' and P'' have this form, so they are code places.

Next, by the definition of the f_r, it is easy to see that

$$\psi_i(w) = \sum_{l=0}^{q^{n/k}-1} \psi_{i-1}(w^{(l)}) \left(g_l[(i-1)(n/k)]\right).$$

The claimed equation follows immediately. □

Observe that $w(P)$ looks like an element of a product matrix. We can write down an explicit matrix product as follows.

Fix $i \in [1, k]$ and $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$. Let $\mathcal{P}_\alpha$ denote the set of code places P of $F_{i(n/k)}$ for which $x_{(i-1)(n/k)}(P) = \alpha$, let $\mathcal{P}'_\alpha$ denote the set of code places Q of $F_{(i-1)(n/k)}$ such that $x_{(i-1)(n/k)}(Q) = \alpha$, and let $\mathcal{P}''_\alpha$ denote the set of code place R of $F_{n/k}$ such that $x_0(R) = \alpha$. Then it is easy to see that for any $Q \in \mathcal{P}'_\alpha$ and $R \in \mathcal{P}''_\alpha$, there is a unique place $P \in \mathcal{P}_\alpha$ such that $P' = Q$ and $P'' = R$, where P' and P'' are as in the above proposition. Conversely, if $P \in \mathcal{P}_\alpha$, then $P' \in \mathcal{P}'_\alpha$ and $P'' \in \mathcal{P}''_\alpha$.

Easily $|\mathcal{P}'_\alpha| = q^{i(n/k)}$ and $|\mathcal{P}''_\alpha| = q^{n/k}$. Let $Q_1^\alpha, \cdots, Q_{q^{i(n/k)}}^\alpha$ be an enumeration of $\mathcal{P}'_\alpha$, and let $R_1^\alpha, \cdots, R_{q^{n/k}}^\alpha$ be an enumeration of $\mathcal{P}''_\alpha$.

Proposition 5.2. *Let $i \in [1,k]$, and let $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$. Write $w = \sum_{l=0}^{q^{n/k}-1} \iota_l(w^{(l)})$ as in Proposition 5.1. For each $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$, define a matrix A^α of shape $q^{(i-1)(n/k)} \times q^{n/k}$ and a matrix B^α of shape $q^{n/k} \times q^{n/k}$ by*

$$A^\alpha_{st} = w^{(t-1)}(Q^\alpha_s) \qquad\qquad B^\alpha_{st} = g_{s-1}(R^\alpha_t).$$

Then for every code place P of $F_{i(n/k)}$, letting $\alpha = x_{(i-1)(n/k)}(P)$ and letting s, t be such that $P' = Q^\alpha_s$ and $P'' = R^\alpha_t$, we have $w(P) = (A^\alpha B^\alpha)_{st}$.

Proof. This is just a restatement of Proposition 5.1. □

Using this proposition, it is not too difficult to define an algorithm Matrix-Encode which encodes $v \in \mathbb{F}_{q^2}^{q^n}$ using a series of $k(q^2-q)$ matrix multiplications of shape

$$\left(q^{n-n/k} \times q^{n/k}\right) \times \left(q^{n/k} \times q^{n/k}\right). \tag{4}$$

We give a careful definition in the appendix (Algorithm 1).

Encoding for messages of length longer than q^n: To encode $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$, we just need to make $q^2-(k+1)q$ calls to Matrix-Encode, using the fact that $f_{sq^n+t} = x_0^s f_t$.

5.3. **Complexity of the Encoding Algorithm.** Performing the pre-computation using the algorithm in [15] requires at most $(n/k)^3 q^{3n/k+4}$ multiplications and divisions in $\mathbb{F}_{q^2}$. Storing the evaluations requires space $O(q^{2n/k+2})$, since there are $q^{n/k}$ functions g_s and $O(q^{n/k+2})$ code places of $F_{n/k}$.

Next, we compute the runtime of the encoding function for general $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$. The runtime of each call to Matrix-Encode is just the runtime of $k(q^2-q)$ matrix multiplications of shape (4). Then the runtime to encode general $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$ is the runtime of $k(q^2-q)^2 = O(kq^4)$ such multiplications. Thus using fast square matrix multiplication, we can encode v using

$$O\left(kq^4 q^{n-2n/k}(q^{n/k})^\omega\right) = O\left(kq^4 (q^n)^{1+\frac{\omega-2}{k}}\right)$$

operations over $\mathbb{F}_{q^2}$, where ω is the exponent of (square) matrix multiplication. Taking $k=2$, this proves the encoding part of Theorem 1.1 by noting that both q and k are constants in the statement, and $N = (q^2-q)q^n$. Using the best known bound $\omega \le 2.37$ [13], we attain the runtime $O\left(kq^4(q^n)^{1+0.37/k}\right)$. When $k \ge 3$, we can instead use fast $(M^2 \times M) \times (M \times M)$ rectangular matrix multiplication; letting ω' be the exponent of such multiplication, we get an algorithm running in time $O(kq^4(q^n)^{1+(\omega'-3)/k})$. We compare the parameters for $k = 2$ and 3 in Table 1 below.

k	2	3
Preprocessing time	$O(N^{1.5}\log^3 N)$	$O(N\log^3 N)$
Encoding time with $\omega = 2$	$O(N)$	$O(N)$
Encoding time with $\omega \approx 2.37$	$O(N^{1.19})$	$O(N^{1.13})$
Encoding time with $\omega' \approx 3.34$	N/A	$O(N^{1.12})$
Smallest q beating Gilbert-Varshamov bound	19	32

Table 1: Comparison of encoding times and code quality for $k = 2, 3$. Here N is the code length, ω is the exponent of square matrix multiplication, and ω' is the exponent of $(M^2 \times M) \times (M \times M)$ rectangular matrix multiplication. In the runtimes, we treat q as a constant.

6. Appendix: Algorithm Matrix-Encode

procedure Matrix-Encode($v \in \mathbb{F}_{q^2}^{q^n}$)
    $W_k \leftarrow \{v\}$
    for i from k to 1 **do**
        $W_{i-1} \leftarrow \emptyset$
        for $w \in W_i$ **do**
            Write $w = \sum_{l=0}^{q^{n/k}-1} \iota_l(w^{(l)})$ as in Proposition 5.1
            Add all $w^{(l)}$ to W_{i-1}
        end for
    end for
    for i from 1 to k **do**
        for $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$ **do**
            for $w \in W_i$ **do**
                Construct the matrix A_w^α corresponding to w in Proposition 5.2, using the $w^{(l)}(Q)$ computed in iteration $i-1$ (when $i = 1$, just use the values of the scalars $w^{(l)}$)
            end for
            Let $\bar{A}^\alpha$ be the matrix made of all A_w^α stacked vertically, and let B^α be as in Proposition 5.2
            Multiply $\bar{A}^\alpha$ by B^α, thus computing $w(P)$ for all $w \in W_i$ and code places P of $F_{i(n/k)}$
        end for
    end for
end procedure

Algorithm 1: The algorithm Matrix-Encode. It inputs $v \in \mathbb{F}_{q^2}^{q^n}$ and outputs $\mathrm{ev}(\psi(v))$.

7. Appendix: Fast Decoding Algorithms

Reed-Solomon codes are widespread in practice partly due to fast algebraic decoding algorithms: the Gorenstein-Zierler decoder, rational approximation using the Euclidean algorithm, the Berlekamp-Massey algorithm and fast Fourier decoders, to name a few. In particular, Reed-Solomon codes can be uniquely decoded in linear time up to the unique decoding limit. In a breakthrough, Sudan designed an algorithm to list decode Reed-Solomon codes beyond half the minimum distance [18]. List decoding is a relaxation of unique decoding where the decoder is allowed to output a list of messages and is deemed successful if the message sent is in the list. Shokrollahi and Wasserman soon generalized Sudan's algorithm to algebraic geometry codes [14]. Shortly thereafter, Guruswami and Sudan designed list decoders for both Reed-Solomon codes and algebraic geometry codes that improved on the error correction of previously known algorithms [8]. A novelty they introduced was to use multiplicities in the interpolation step.

We present a unique decoding algorithm that corrects a fraction of errors close to half the relative distance and a list decoding algorithm to correct beyond that. The algorithms are presented as specializations of the Shokrollahi-Wasserman algorithm to the Garcia-Stichtenoth towers. In particular, we obtain the unique decoding algorithm as a special case of the list decoding algorithm. The runtime of both algorithms is dominated by the interpolation step which, when performed using generic linear algebra, has exponent ω. We briefly sketch the main ideas in list decoding leaving the details to a subsequent longer version of the paper.

7.1. **Modified Shokrollahi-Wasserman Algorithm.** Consider codes of block length $N = q^n(q^2 - q)$ and dimension K constructed in §4 with parameter k. Let $y = (y_1, y_2, \ldots, y_N) \in \mathbb{F}_{q^2}^N$ denote the received word. Let b be a bound on the number of messages allowed in the list. Let β be an agreement parameter (determined later) such that $N - \beta - 1$ is the number of errors up to which we need to correct. The algorithm first interpolates a nonzero polynomial

$$H(T) := u_b T^b + u_{b-1} T^{b-1} + u_1 T + u_0 \in F_n[T]$$

in an indeterminate T such that every message in the list (that is, every message whose encoding agrees with the received word in at least $\beta + 1$ places) is a root of $H(T)$. Then a root finding algorithm is used to enumerate the roots of $H(T)$ that are in the message Riemann-Roch space $\mathcal{L}(G + KP_\infty^{(n)})$ as a list of candidate messages. The encoding algorithm is then used to check which messages sufficeintly agree with the received word and indeed belong in the list.

Interpolation: To compute such a polynomial $H(T)$, we require

$$H(y_i)(P_i) = \left(\sum_{j=0}^{b} u_j(P_i) y_i^j \right) = 0, \forall i \in \{1, 2, \ldots, N\} \tag{5}$$

and that

$$u_j \in \mathcal{L}(G + w_j P_\infty), \forall j \in \{0, 1, \ldots, b\} \tag{6}$$

where $w_j := \beta - Kj - (j+1)q^n(kq + k - 1)$. The latter constraint $u_j \in \mathcal{L}(G + w_j P_\infty)$ ensures that when we substitute a function $f \in \mathcal{L}(G + KP_\infty)$, the resulting function $H(f)$ has a pole divisor of degree at most β.

We claim that there is a nonzero polynomial $H(T)$ satisfying the constraints. Recall that the sequence of functions $f_0, f_1, \ldots$ constructed in §3 satisfy $f_0, f_1, \ldots, f_{w_j} \in \mathcal{L}(G + w_j P_\infty)$ for all $j \in \{0, 1, \ldots, b\}$. We enforce the second constraint (equation 6) by constraining u_j to be in the span of $\{f_0, f_1, \ldots, f_{w_j}\}$. In particular, writing each u_j as an unknown linear combination of $\{f_0, f_1, \ldots, f_{w_j}\}$, the first constraint (equation 5) is an $\mathbb{F}_{q^2}$-linear system in $w_0 + w_1 + \ldots + w_b + b$ variables. Choose β to be large enough to ensure the number of variables exceeds the number of constraints N.

Say $f \in \mathcal{L}(G + KP_\infty^{(n)})$ agrees with $(y_0, y_1, \ldots, y_N)$ at more than β places. Then $H(f)$ has a zero at more than β places yet pole degree at most β. Hence $H(f) = 0$ and f is indeed a root of $H(T)$.

Pre-processing: Compute and store the evaluations of $f_0, f_1, \ldots, f_{w_0}$ at the evaluation places. A straightforward way to implement this step is to use the encoding algorithm once for each $f_0, f_1, \ldots, f_{w_0}$. This involves $w_0 \leq N$ encodings and in total takes $O(N^{1+\omega/2})$ time. The storage required is clearly linear in N.

Fast Unique Decoding: We obtain our unique decoding algorithm by choosing $b = 1$. In particular, since $H(T)$ is now a linear polynomial, the root finding step is trivial and involves just one division of functions. We next develop a fast algorithm for the interpolation step and analyze the error correction capability.

Given the received word and the stored evaluations of $f_0, f_1, \ldots, f_{w_0}$ at the evaluation places, the linear system in equation 5 can be written down immediately in $O(N^2)$ time. All that remains is to solve the linear system and obtain $H(T)$. This has runtime exponent ω using generic linear algebra. The error correction capability is determined by how large β must be to ensure that the linear system has a nonzero solution. For $b = 1$, we may choose $\beta = \frac{1}{2}(N + K) + \frac{3}{2}q^n(kq + k - 1)$, allowing us to correct $\frac{1}{2}(N - K - 1) - \frac{3}{2}q^n(kq + k - 1)$ errors. This is close to half the designed distance of the code, since $\frac{3}{2}q^n(kq + k - 1)/N = \frac{3}{2}(kq + k - 1)/(q^2 - q)$ is only $\frac{3}{2}$ times our bound on $1-(R+\delta)$. There are ways to decrease the term $\frac{3}{2}q^n(kq+k-1)/N$ so as to correct closer to half the designed distance. We refrain from detailing the changes since the list decoding algorithm will allow us to correct beyond half the designed distance.

Root Finding for List Decoding using Hensel Lifting: Allowing for large b lets us choose a β small enough to list decode well beyond half the designed distance. The error correction guarantee is close to that obtained by Shokrollahi-Wasserman, worse only by a small term determined by k.

The main algorithmic challenge is root finding: to enumerate all the roots of $H(Y)$ in $\mathcal{L}(G+KP_\infty)$. We employ the Hensel lifting algorithm of Augot and Pecquet to perform root finding [2]. Their algorithm reduces $H(T)$ at every evaluation place and finds roots in the respective residue fields. Using Hensel lifting at each evaluation place, each of these roots is lifted to a root modulo a high enough power of the place. Eventually roots modulo high powers of evaluation places are lifted to functions in the message Riemann-Roch space. The main algorithmic ingredient required in the Augot-Pecquet algorithm is a means to evaluate basis functions of the message space and coefficients of $H(T)$ at the evaluation places. This we can accomplish efficiently in time quadratic in N using our encoding algorithm and pre-processing.

References

[1] I. Aleshnikov, P. V. Kumar, K. W. Shum, and H. Stichtenoth. On the splitting of places in a tower of function fields meeting the Drinfeld-Vlăduţ bound. *IEEE Transactions on Information Theory*, 47(4):1613–1619, May 2001.

[2] Daniel Augot and Lancelot Pecquet. A Hensel lifting to replace factorization in list-decoding of algebraic-geometric and Reed-Solomon codes. *IEEE Transactions on Information Theory*, 46:2605–2614, 2000.

[3] V. G. Drinfeld and S. G. Vlăduţ. The number of points of an algebraic curve. *Func. Anal.*, 17:53–54, 1983.

[4] Erez Druk and Yuval Ishai. Linear-time encodable codes meeting the Gilbert-Varshamov bound and their cryptographic applications. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 169–182, New York, NY, USA, 2014. ACM.

[5] Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of Number Theory*, 61(2):248 – 273, 1996.

[6] E. N. Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, May 1952.

[7] V. D. Goppa. Codes on algebraic curves. *Soviet Math. Dokl.*, 24(1):170–172, 1981.

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

[9] V. Guruswami and M. Sudan. On representations of algebraic-geometry codes. *IEEE Transactions on Information Theory*, 47(4):1610–1613, May 2001.

[10] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51-10:3393–3400, 2005.

[11] F. Hess. Computing Riemann-Roch spaces in algebraic function fields and related topics. *Journal of Symbolic Computation*, 33(4):425 – 445, 2002.

[12] M. D. Huang and D. Ierardi. Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 678–687, Oct 1991.

[13] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[14] M. A. Shokrollahi and H. Wasserman. List decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(2):432–437, Mar 1999.

[15] K. W. 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, Sep 2001.

[16] Kenneth Shum. *A Low-Complexity Construction of Algebraic Geometric Codes Better Than the Gilbert-Varshamov Bound*. PhD thesis, University of Southern California, 12 2000.

[17] Daniel Speilman. Linear time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 46-6:1723–1731, 1996.

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

[19] M. A. Tsfasman, S. G. Vlăduţ, and T. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982.

[20] R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Dokl. Acad. Nauk SSSR*, 117:739–741, 1957.