12-2009

DECODING OF MULTIPOINT ALGEBRAIC GEOMETRY CODES VIA LISTS

Nathan Drake
Clemson University, nathan.drake@ngu.edu

DECODING OF MULTIPOINT ALGEBRAIC GEOMETRY CODES VIA LISTS

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematical Sciences

by
Nathan Drake
December 2009

Dr. Gretchen Matthews, Committee Chair
Dr. Neil Calkin
Dr. Shuhong Gao
Dr. Hui Xue

ABSTRACT

Algebraic geometry codes have been studied greatly since their introduction by Goppa [12]. Early study had focused on algebraic geometry codes $C_{\mathcal{L}}(D, G)$ where G was taken to be a multiple of a single point. However, it has been shown that if we allow G to be supported by more points, then the associated code may have better parameters. We call such a code a multipoint code and if G is supported by m points, then we call it an m-point code. In this dissertation, we wish to develop a decoding algorithm for multipoint codes. We show how we can embed a multipoint algebraic geometry code into a one-point supercode so that we can perform list decoding in the supercode. From the output list, we determine which of the elements is a codeword in the multipoint code. In this way we have unique decoding up to the minimum distance for multipoint algebraic geometry codes, provided the parameters of the list decoding algorithm are set appropriately.

ACKNOWLEDGMENTS

I appreciate the supervision, encouragement, and expertise provided to me by my advisor Dr. Gretchen Matthews. I am indebted to her for the continuous support and patience throughout my graduate career.

I am thankful to Dr. N. Calkin, S. Gao, and H. Xue for serving on my dissertation committee and for their help and advice throughout my graduate education. Also I would like to thank the Department of Mathematical Sciences for their continuous support.

Finally I would like to thank my parents for their support and guidance. And to my wife Mandy, she put her dreams on hold for mine. I could not have done this without her love and encouragement.

TABLE OF CONTENTS

Page

TITLE PAGE i

ABSTRACT iii

ACKNOWLEDGMENTS v

CHAPTER

1. INTRODUCTION 1

2. BACKGROUND 4

2.1 Coding Theory 4
2.2 Algebraic Function Fields and Curves 6
2.3 Algebraic Geometry Codes 26

3. LIST DECODING 31

3.1 List Decoding Reed-Solomon Codes 31
3.1.1 Sudan List Decoding Algorithm for Reed-Solomon Codes . . 32
3.1.2 Guruswami-Sudan List Decoding Algorithm for Reed-Solomon Codes 34
3.2 List Decoding AG Codes 36
3.2.1 Shokrollahi-Wasserman List Decoding Algorithm for AG Codes 37
3.2.2 Guruswami-Sudan List Decoding Algorithm for One-point Codes 39
3.3 Improved Bounds on the List Size in the Guruswami-Sudan Algorithm for One-point Codes 42
3.4 Examples 47
3.5 List Decoding AG Codes Over Rings 48
3.6 List Decoding AG Codes Using Places of Higher Degree 51

4. DECODING GENERAL AG CODES USING LISTS 56

4.1 Introduction 56
4.2 A Minimum Distance Decoder for Multipoint Codes Via Lists 57
4.3 A Minimum Distance Decoder for Multipoint Codes Using Lists, Multiple Embeddings, and GCD 62
4.4 Examples 64

5. CONCLUSION 67

Table of Contents (Continued)

Page

INDEX 68

BIBLIOGRAPHY 70

CHAPTER 1

INTRODUCTION

A one-point algebraic geometry code is built upon a divisor G supported by a single place and another divisor D. These codes have been studied in great detail, including determining code parameters as well as developing decoding algorithms. Among the decoding algorithms are list decoding algorithms which return lists of possible sent words (meaning those within a fixed radius of the sent word) rather than just a single codeword (meaning the nearest codeword). These algorithms allow for decoding beyond the normal decoding radius in some cases.

An m-point algebraic geometry code is also a code built upon a divisor G' and another divisor D', but in this case the divisor G' is supported by m places rather then a single place. If $m \geq 2$, then the code is called a multipoint code. The parameters of multipoint codes can be better than those of comparable one-point code. However decoding algorithms for these m-point codes are lacking. In particular, known algorithms [7] and [30] only decode up to the designed distance or the order bound on the minimum distance. In this thesis we show how to embed a multipoint code into a one-point code. We are then able to leverage known list decoding algorithms and decode in the one-point supercode. We then determine which of the words in the list are also codewords in the multipoint code. If the parameters in the list decoding algorithm are appropriately set, then we obtain a unique codeword in the multipoint code. Since we go through the list to determine which codewords are in the multipoint code, we would like to have a better idea on the size of the list. Thus, we provide better parameter choices for the Guruswami-Sudan list decoding algorithm that give a better bound on the list size.

This thesis is arranged as follows.

Chapter 2 details the necessary background for coding theory. In this chapter we set up the necessary notation and terminology. The background considers linear codes and then narrows to focus on algebraic geometry codes.

Chapter 3 details some of the breakthroughs in list decoding. We begin with a focus on the seminal list decoding algorithm for Reed-Solomon codes. We then move on to the list decoding algorithms developed for algebraic geometry codes. Next we show how we can choose parameters in the Guruswami-Sudan list decoding algorithm for algebraic geometry codes to obtain a better bound on the list size. We obtain an analogous result in the following section for decoding algebraic geometry codes over rings. We close the chapter by developing a list decoding algorithm for an algebraic geometry code $C_{\mathcal{L}}(D, \alpha P)$ where P is taken to be a place of degee greater than one.

Chapter 4 details the how list decoding may be used to decode multipoint codes. We begin by showing how to embed a multipoint code in a one-point code. We then discuss how to use list decoding in the one-point supercode to decode the multipoint code. We follow that by discussing various ways to determine the sent word for the multipoint code. We also discuss the use of multiple embeddings and greatest common divisors for decoding multipoint codes. Chapter 4 concludes with examples.

Chapter 5 concludes the thesis with open problems for future work.

CHAPTER 2

BACKGROUND

This chapter introduces the notation and terminology used throughout this dissertation. A brief introduction to algebraic function fields and curves, coding theory, and algebraic geometry codes is provided. Necessary details on each of these subjects can be found in [33].

2.1 Coding Theory

We begin with a brief discussion of some basic coding theory concepts. A nice overview of coding theory can be found in [21].

Given a power q of a prime number, $\mathbb{F}_q$ denotes the field with q elements. Given a field $\mathbb{F}$, $\mathbb{F}[x]$ denotes the polynomial ring over $\mathbb{F}$ in the indeterminate x and $\mathbb{F}(x)$ denotes the rational function field over $\mathbb{F}$. The set of integers is denoted by $\mathbb{Z}$.

Definition 2.1. *A linear code C of length n over $\mathbb{F}_q$ is a linear subspace of $\mathbb{F}_q^n$. Elements of the subspace C are called codewords of C.*

Since we only consider linear codes in this thesis, we say code to mean linear code.

Definition 2.2. *Given a code over $\mathbb{F}_q$, we say that $\mathbb{F}_q$ is the alphabet for C.*

Definition 2.3. *The dimension of a code C over $\mathbb{F}_q$ is $dim_{\mathbb{F}_q} C$, the dimension of C as a vector space over $\mathbb{F}_q$.*

Given a vector $w \in \mathbb{F}_q^n$, w_i denotes the i^{th} coordinate of w. We sometimes refer to such a vector as a word in $\mathbb{F}_q^n$.

Definition 2.4. *The Hamming distance between two vectors w and w' in $\mathbb{F}_q^n$ is the number of coordinates in which they differ; that is,*

$$d(w, w') = |\{i : w_i \neq w'_i\}|.$$

Definition 2.5. *The weight of a codeword is its number of nonzero coordinates.*

Definition 2.6. *The minimum distance of a code C is*

$$d = min\{d(c, c') : c, c' \in C, c \neq c'\},$$

the smallest of the distances between distinct codewords.

We often write $d(C)$ to denote the minimum distance of a code C.

For a linear code C, it can be shown that the minimum distance is equal to the minimum weight of a nonzero codeword in C.

Definition 2.7. *A Hamming sphere of radius r about $w \in \mathbb{F}^n$ is $\{y \in \mathbb{F}^n : d(w, y) \leq r\}$.*

Definition 2.8. *An $[n, k, d]$ code is a code of length n, dimension k, and minimum distance d. An $[n, k]$ code is a code of length n and dimension k. An $[n, k, \geq d]$ code is a code of length n, dimension k, and minimum distance at least d.*

Remark 2.9. *Suppose C is an $[n, k, d]$ code over $\mathbb{F}_q$. Then the Hamming spheres of radius $\frac{d-1}{2}$ about the codewords of C are disjoint. Hence, if a word in $\mathbb{F}_q^n$ differs from a codeword in $\left\lfloor \frac{d-1}{2} \right\rfloor$ or fewer coordinates, it lies in exactly one Hamming sphere about a codeword. For this reason, it is said that C can correct $\left\lfloor \frac{d-1}{2} \right\rfloor$ or fewer errors.*

Example 2.10. *Let q be a power of a prime number and k be chosen such that $1 \leq k \leq q - 1$. Let the nonzero elements of $\mathbb{F}_q$ be denoted $\{x_1, x_2, \ldots, x_{q-1}\}$. Then a code C can be created by evaluating every polynomial in $F_q[x]$ of degree less than k at each nonzero field element.*

More precisely,

$$C = \{(f(x_1), f(x_2), \ldots, f(x_{q-1})) : f \in \mathbb{F}_q[x], \deg(f) < k\} \subseteq \mathbb{F}_q^{q-1}$$

is a code over $\mathbb{F}_q$. In fact, C is a $[q-1, k, q-k]$ code called a Reed-Solomon code. As this example shows, the length of a Reed-Solomon code over $\mathbb{F}_q$ is determined by the size of the field $\mathbb{F}_q$. Such a code is considered short relative to the alphabet size.

Definition 2.11. *Let $a, b \in \mathbb{F}_q^n$. The inner product of a and b is defined by*

$$< a, b >:= \sum_{i=1}^{n} a_i b_i \in \mathbb{F}_q.$$

Definition 2.12. *If $C \subseteq \mathbb{F}_q^n$ is a code, then*

$$C^{\perp} := \{u \in \mathbb{F}_q^n :< u, c >= 0 \textit{ for all } c \in C\}$$

is called the dual of C.

Given an $[n, k]$ code C over $\mathbb{F}_q$, $C^{\perp}$ is an $[n, n-k]$ code over $\mathbb{F}_q$.

2.2 Algebraic Function Fields and Curves

We wish to describe algebraic geometry codes, a generalization of the Reed-Solomon codes that were introduced in Example 2.10. However, we must develop the necessary background on algebraic function fields in order to do so. Stichtenoth [33] provides an in-depth look at this topic. Hence, many of the definitions and results from this section are taken from [33].

Definition 2.13. *An algebraic function field F of one variable over a field K is an extension field F of K such that F is a finite algebraic extension of $K(x)$ for some element $x \in F$ which is transcendental over K.*

For convenience, we sometimes refer to an algebraic function field as a function field for short. We use the notation F/K to mean that F is a function field over K.

Example 2.14. *Let K be a field and x be transcendental over F. The field*

$$K(x) := \left\{ \frac{f(x)}{g(x)} : f(x), g(x) \in K[x], g(x) \neq 0 \right\}$$

is a transcendental extension of K. Then clearly $K(x)$ is a finite extension of $K(x)$. This can be viewed as

$$\begin{tikzcd}
K(x) \arrow[d, dash] \\
K(x) \arrow[d, dash] \\
K.
\end{tikzcd}$$

Thus, $K(x)/K$ is an algebraic function field, called the rational function field.

Example 2.15. *Let q be a prime power. Let $K = \mathbb{F}_{q^2}$ and x be so that $\mathbb{F}_{q^2}(x)$ is transcendental over $\mathbb{F}_{q^2}$. Let y be such that*

$$y^q + y = x^{q+1}.$$

Then $\mathbb{F}_{q^2}(x, y)$ is a finite algebraic extension of $\mathbb{F}_{q^2}(x)$. This can be pictured as

$$\begin{tikzcd}
\mathbb{F}_{q^2}(x, y) \arrow[d, dash] \\
\mathbb{F}_{q^2}(x) \arrow[d, dash] \\
\mathbb{F}_{q^2}.
\end{tikzcd}$$

Thus, $\mathbb{F}_{q^2}(x, y)/\mathbb{F}_{q^2}$ is an algebraic function field called the Hermitian function field.

Definition 2.16. *A valuation ring of a function field F/K is a ring $\mathcal{O} \subseteq F$ with the following properties:*

1. *$K \subsetneqq \mathcal{O} \subsetneqq F$, and*
2. *for any $z \in F$, $z \in \mathcal{O}$ or $z^{-1} \in \mathcal{O}$.*

Definition 2.17. *A place P of a function field F/K is a maximal ideal of some valuation ring $\mathcal{O}$ of F/K. The set of places of F/K is denoted $\mathbb{P}_F$.*

Given a place P, we often use the notation $\mathcal{O}_P$ for the ring that has P as its maximal ideal. Note that the ring $\mathcal{O}_P$ is uniquely determined by P because for $x \in F \setminus \{0\}$, $x \in P$ if and only if $x^{-1} \notin \mathcal{O}$.

Definition 2.18. *Given a place P of a function field F/K, any function $t \in P$ such that $P = t\mathcal{O}$ is called a local parameter for P.*

Definition 2.19. *Let F/K be a function field and $P \in \mathbb{P}_F$.*

1. *The field $F_P := \mathcal{O}_P/P$ is the residue class field of P.*
2. *The map*
$$\begin{array}{ccc} F & \rightarrow & F_P \cup \{\infty\} \\ x & \mapsto & x(P) \end{array}$$
is called the residue class map with respect to P where
   - *for $x \in \mathcal{O}_P$, the residue class of x modulo P is $x(P) \in \mathcal{O}_P/P$, and*
   - *for $x \in F \setminus \mathcal{O}_P, x(P) := \infty$.*
3. *The degree of the place P is deg $P := [F_P : K]$.*

Notice that the residue class map gives an embedding of K into $\mathcal{O}_P/P$. Thus we consider K as a subfield of $\mathcal{O}_P/P$ by way of this embedding. This allows us to define the degree of a place as above.

Example 2.20. *Consider the Hermitian function field*

$$H = \mathbb{F}_{q^2}(x, y) \text{ over } \mathbb{F}_{q^2} \text{ where } y^q + y = x^{q+1}.$$

This function field has $q^3 + 1$ places of degree one. For each pair in $(\alpha, \beta) \in \mathbb{F}_{q^2} \times \mathbb{F}_{q^2}$ with $\beta^q + \beta = \alpha^{q+1}$, there is a unique place $P_{\alpha\beta} \in \mathbb{P}_H$ of degree one such that $x(P_{\alpha\beta}) = \alpha$ and $y(P_{\alpha\beta}) = \beta$. In particular, for all $\alpha \in \mathbb{F}_{q^2}$ there exist q distinct elements $\beta \in \mathbb{F}_{q^2}$ with $\beta^q + \beta = \alpha^{q+1}$. Therefore, the number of such places $P_{\alpha\beta}$ is q^3. In addition to these q^3 places of degree one, there is also one place at infinity, P_∞, and $\deg P_\infty = 1$. Hence the Hermitian function field over F_{q^2} has $q^3 + 1$ places of degree one.

Definition 2.21. *A discrete valuation of a function field F/K is a function*

$$v : F \to \mathbb{Z} \cup \{\infty\}$$

with the following properties:

1. *$v(x) = \infty$ if and only if $x = 0$.*
2. *$v(xy) = v(x) + v(y)$ for any $x, y \in \mathbb{F}$.*
3. *$v(x + y) \geq min\{v(x), v(y)\}$ for any $x, y \in \mathbb{F}$.*
4. *There exists an element $z \in \mathbb{F}$ with $v(z) = 1$.*
5. *$v(a) = 0$ for any $a \in K \setminus \{0\}$.*

Definition 2.22. *To any place $P \in \mathbb{P}_F$ we associate a function $v_P : F \to \mathbb{Z} \cup \{\infty\}$ that is a discrete valuation of F/K. To do so, choose a local parameter t for P. Let $\mathcal{O}_P^*$ denote the group of units of $\mathcal{O}_P$. Then every $f \in F \setminus \{0\}$ has a unique representation $f = t^n u$ with $u \in \mathcal{O}_P^*$ and $n \in \mathbb{Z}$. Define*

$$v_P(f) := n,$$

and set

$$v_P(0) := \infty.$$

Theorem 2.1. *Let F/K be a function field. The function v_P is a discrete valuation of F/K for any place $P \in \mathbb{P}_F$. Additionally, we have*

$$\begin{aligned} \mathcal{O}_P &= \{f \in F | v_P(f) \geq 0\}, \\ \mathcal{O}_P^* &= \{f \in F | v_P(f) = 0\}, \text{ and} \\ P &= \{f \in F | v_P(f) > 0\}. \end{aligned}$$

Definition 2.23. *Let $z \in F$ and $P \in \mathbb{P}_F$ where F/K is a function field. A place P is a zero of z of order m if and only if*

$$v_p(z) = m > 0.$$

A place P is a pole of order m of z if and only if

$$v_p(z) = -m < 0.$$

Definition 2.24. *The free abelian group generated by the places of a function field F/K, denoted $\mathcal{D}_F$, is called the divisor group of F/K. The elements of $\mathcal{D}_F$ are called divisors of F/K. In other words, a divisor is a formal sum*

$$D = \sum_{P \in \mathbb{P}_F} n_P P \text{ with } n_P \in \mathbb{Z}, \text{ almost all } n_P = 0.$$

The support of the divisor D is

$$\textit{supp } D := \{P \in \mathbb{P}_F \ : \ n_P \neq 0\}.$$

Example 2.25. *Let $P \in \mathbb{P}_F$. Then $\frac{1}{2}P$ is not a divisor because $\frac{1}{2} \notin \mathbb{Z}$.*
By Corollary I.3.2 [33], any function field has infinitely many places. Hence, $\sum_{P \in \mathbb{P}_F} P$ is not a divisor because the sum is not finite.
Let $P_1, P_2, \ldots, P_m, P_\infty \in \mathbb{P}_F$ where m is a positive integer. Then $A = \sum_{i=1}^m P_i - mP_\infty$ is a divisor.

Given divisors $A = \alpha_1 P_1 + \cdots + \alpha_m P_m$, $B = \beta_1 P_1 + \cdots + \beta_m P_m \in \mathcal{D}_F$, we say that

$$A \leq B$$

if and only if

$$\alpha_i \leq \beta_i \text{ for all } i, 1 \leq i \leq m.$$

Example 2.26. *Let $P_1, P_2, \ldots, P_m, P_\infty \in \mathbb{P}_F$ where m is a positive integer. Let A and B be divisors of F/K such that $A = -mP_\infty$ and $B = \sum_{i=1}^m P_i - mP_\infty$. Then $A \leq B$.*

Definition 2.27. *The degree of a divisor $A = \alpha_1 P_1 + \cdots + \alpha_m P_m \in \mathcal{D}_F$ is*

$$deg(A) = \sum_{i=1}^{m} \alpha_i \cdot deg(P_i).$$

Example 2.28. *Let $P \in \mathbb{P}_F$ be a place of degree one and $B = -mP$. Then*

$$deg(B) = -m.$$

Let $P_1, P_2, \ldots, P_m, P_\infty \in \mathbb{P}_F$ be places of degree one where m is a positive integer and $A = \sum_{i=1}^m P_i - mP_\infty$. Then

$$deg(A) = 0.$$

Next, we consider an important class of divisors of degree zero, called principal divisors [33, Definition I.4.1].

Definition 2.29. *Let $x \in F \setminus \{0\}$ and denote by Z (resp. N) the set of zeros (resp. poles) of x in $\mathbb{P}_F$. Then we define*

$(x)_0 \quad := \quad \sum_{P \in Z} v_p(x)P,$ *the zero divisor of x;*

$(x)_\infty \quad := \quad \sum_{P \in N} (-v_p(x))P,$ *the pole divisor of x; and*

$(x) \quad := \quad (x)_0 - (x)_\infty,$ *the principal divisor of x.*

Note that

$$(x) = \sum_{P \in \mathbb{P}_F} v_p(x)P.$$

It can be shown that

$$\deg\left(\sum_{P \in Z} v_p(x)\right) = \deg\left(\sum_{P \in N} v_p(x)\right).$$

(See [33, Theorem I.4.11].) Therefore, every principal divisor has degree zero. However, as we will see, not every divisor of degree zero is a principal divisor.

Given a function $f \in F$, the principal divisor (f) is called the divisor of f.

Proposition 2.30. *Consider a function field F/K. Let $f, h \in F \setminus \{0\}$. Then,*

1. $(f \cdot h) = (f) + (h)$.
2. $(f^{-1}) = -(f)$.
3. $\left(\frac{f}{h}\right) = (f) - (h)$.

Proof. Let $f, h \in F \setminus \{0\}$. Then,

$$\begin{aligned} (f \cdot h) &= \textstyle\sum_{P \in \mathbb{P}_F} v_P(f \cdot h) \\ &= \textstyle\sum_{P \in \mathbb{P}_F} (v_P(f) + v_P(h)) \\ &= \textstyle\sum_{P \in \mathbb{P}_F} v_P(f) + \sum_{P \in \mathbb{P}_F} v_P(h) \\ &= (f) + (h). \end{aligned}$$

Therefore, $(f \cdot h) = (f) + (h)$.

To determine (f^{-1}), note that $f \cdot f^{-1} = 1$. By Definition 2.21 (5)

$$(1) = \sum_{P \in \mathbb{P}_F} v_P(1) = 0.$$

Thus,

$$(f \cdot f^{-1}) = 0.$$

Now from the previous result, we have

$$(f \cdot f^{-1}) = (f) + (f^{-1}).$$

Combining these, we see that

$$(f^{-1}) = -(f).$$

Finally, we consider $\left(\frac{f}{h}\right)$. From the previous result we have

$$\left(\frac{f}{h}\right) = \left(f \cdot h^{-1}\right).$$

Then, from the first result, we have

$$\left(f \cdot h^{-1}\right) = (f) + \left(h^{-1}\right) = (f) - (h).$$

$\square$

Example 2.31. *Consider the Hermitian function field $\mathbb{F}_{q^2}(x, y)$ where $y^q + y = x^{q+1}$.*

Let $a \in \mathbb{F}_{q^2}$. The divisor of $x - a$ is

$$(x - a) = \sum_{\substack{b \in \mathbb{F}_{q^2} \\ b^q + b = a^{q+1}}} P_{ab} - qP_\infty.$$

The divisor of y is

$$(y) = (q + 1)P_{00} - (q + 1)P_\infty.$$

If $b \in \mathbb{F}_{q^2}$ and $b^q + b = 0$, then

$$(y - b) = (q + 1)P_{0b} - (q + 1)P_\infty.$$

Finally, consider the function $y - b$ where $b \in \mathbb{F}_{q^2}$ and $b^q + b \neq 0$. The divisor of $y - b$ is

$$(y - b) = \textstyle\sum_{a \in \mathbb{F}_{q^2},\, b^q + b = a^{q+1}} P_{ab} - (q + 1)P_\infty.$$

Definition 2.32. *For a divisor $A \in \mathcal{D}_F$,*

$$\mathcal{L}(A) := \{x \in F : (x) \geq -A\} \cup \{0\}.$$

We can view this definition in the following manner: if

$$A = \sum_{i=1}^{r} n_i P_i - \sum_{j=1}^{s} m_j Q_j$$

with $n_i, m_j > 0$ then the nonzero elements of $\mathcal{L}(A)$ are $x \in F \setminus \{0\}$ such that

(1) x has a zero of order $\geq m_j$ at Q_j for $j = 1, \ldots, s$, and

(2) x may have poles only at the places $P_1, \ldots, P_r$, with the pole order at P_i being at most n_i for $i = 1, \ldots, r$.

Definition 2.33. *Let A and A' be divisors. The divisors A and A' are equivalent if*

$$A = A' + (x)$$

for some $x \in F \setminus \{0\}$. This is denoted by $A \sim A'$.

Remark 2.34. *Let $A \in \mathcal{D}_F$. Then*

1. *$x \in \mathcal{L}(A)$ if and only if $v_P(x) \geq -v_P(A)$ for all $P \in \mathbb{P}_F$.*
2. *$\mathcal{L}(A) \neq \{0\}$ if and only if there is a divisor $A' \sim A$ with $A' \geq 0$.*

Example 2.35. *Consider $\mathbb{F}_{q^2}(x, y)$ where $y^q + y = x^{q+1}$. Then for $a \in \mathbb{F}_{q^2}$,*

$$(x - a) = \textstyle\sum_{b \in \mathbb{F}_{q^2},\, b^q + b = a^{q+1}} P_{ab} - qP_\infty \geq -qP_\infty.$$

Therefore, $x - a \in \mathcal{L}(qP_\infty)$ for all $a \in \mathbb{F}_{q^2}$.

Now consider the function $y - b$ where $b \in \mathbb{F}_{q^2}$ and $b^q + b \neq 0$. Then

$$(y - b) = \sum_{a \in \mathbb{F}_{q^2},\, b^q + b = a^{q+1}} P_{ab} - (q+1)P_\infty \ngeq -qP_\infty.$$

Therefore, $y - b \notin \mathcal{L}(qP_\infty)$.

Lemma 2.36. *Let F/K be a function field and $A \in \mathcal{D}_F$. Then $\mathcal{L}(A)$ is a vector space over K.*

Proof. Let $x, y \in \mathcal{L}(A) \setminus \{0\}$. Then for any $P \in \mathbb{P}_F$,

$$v_P(x+y) \geq min\{v_p(x), v_p(y)\}$$

by Definition 2.21. Thus

$$v_P(x+y) \geq -v_P(A),$$

and

$$x + y \in \mathcal{L}(A).$$

Let $a \in K \setminus \{0\}$. Then for any $P \in \mathbb{P}_F$,

$$v_P(ax) = v_P(a) + v_P(x) = v_P(x) \geq -v_P(A),$$

and

$$ax \in \mathcal{L}(A).$$

If $a = 0$, then $ax = 0 \in \mathcal{L}(A)$. □

Definition 2.37. *Let A be a divisor of F/K. The dimension of the divisor A is*

$$dim\ A := \dim_K \mathcal{L}(A).$$

We sometimes write $l(A)$ to mean $\dim A$*.*

Lemma 2.38. *Let A, B be divisors of F/K with $A \leq B$. Then we have*

$$\mathcal{L}(A) \subseteq \mathcal{L}(B)$$

and

$$\dim(\mathcal{L}(B)/\mathcal{L}(A)) \leq deg\ B - deg\ A.$$

Proof. Suppose $x \in \mathcal{L}(A) \setminus \{0\}$. Then

$$(x) \geq -A.$$

Now since $A \leq B$,

$$(x) \geq -A \geq -B.$$

Thus

$$x \in \mathcal{L}(B)$$

and

$$\mathcal{L}(A) \subseteq \mathcal{L}(B)$$

since $0 \in \mathcal{L}(B)$.

Assume that $B = A + P$ for some $P \in \mathbb{P}_F$. Now choose an element $t \in F$ such that

$$v_P(t) = v_P(B) = v_P(A) + 1.$$

Then for $x \in \mathcal{L}(B)$ we have

$$v_P(x) \geq -v_P(B).$$

Thus

$$v_P(x) \geq -v_P(t),$$

so

$$xt \in O_P$$

since $v_P(xt) \geq 0$. Therefore we have a K-linear map

$$\begin{array}{rccc} \psi : & \mathcal{L}(B) & \longrightarrow & F_P, \\ & x & \longmapsto & (xt)(P). \end{array}$$

We can see that x is in the kernel of ψ if and only if $v_P(xt) > 0$; that is to say, x is in the kernel of ψ if and only if $v_P(x) \geq -v_P(A)$. Thus

$$Ker(\psi) = \mathcal{L}(A).$$

Therefore ψ induces a K-linear injective mapping from $\mathcal{L}(B)/\mathcal{L}(A)$ to F_P. Therefore,

$$dim(\mathcal{L}(B)/\mathcal{L}(A)) \leq dim\ F_P = deg\ B - deg\ A$$

since $dim\ F_P = deg\ P = deg\ B - deg\ A$. Now the general result follows by repeated application. □

Definition 2.39. *The genus g of F/K is defined by*

$$g := max\{deg\ A - \dim\ A + 1 : A \in \mathcal{D}_F\}.$$

Example 2.40. *It can be shown that the genus of the Hermitian function field over $\mathbb{F}_{q^2}$ is*

$$g = \frac{q(q-1)}{2}.$$

Lemma 2.41.

1. *Let A, A' be divisors with $A \sim A'$. Then we have*

$$dim\ A = dim\ A'$$

and

$$deg\ A = deg\ A'.$$

2. *If $deg\ A < 0$ then $dim\ A = 0$.*

Proof. 1. Let A, A' be divisors such that $A \sim A'$. Then $\mathcal{L}(A)$ and $\mathcal{L}(A')$ are isomorphic as vector spaces. Thus

$$dim\ A = dim\ \mathcal{L}(A) = dim\ \mathcal{L}(A') = dim\ A'.$$

Additionally, $A \sim A'$, so $A = A + (x)$ for some $x \in F \setminus \{0\}$. Thus,

$$\deg A = \deg(A' + (x)) = \deg(A') + \deg((x)).$$

But $\deg((x)) = 0$ since (x) is a principal divisor. Therefore,

$$\deg A = \deg(A').$$

2. Suppose that *dim* $A > 0$. Then there is a divisor $A' \sim A$ such that $A' \geq 0$. Thus,

$$\deg A = \deg A' \geq 0.$$

Therefore, if $\deg A < 0$ then *dim* $A = 0$.

□

Definition 2.42. *Let F/K be a function field of genus g. A divisor W of F/K is called a canonical divisor of F/K if and only if*

$$deg\ W = 2g - 2$$

and

$$dim\ W = g.$$

Theorem 2.2. *(Riemann-Roch Theorem). Let W be a canonical divisor of F/K. Then, for $A \in \mathcal{D}_F$, the dimension of A is*

$$dim\ A = deg\ A + 1 - g + dim(W - A).$$

Proof. See [33, Theorem I.5.15]. □

Corollary 2.43. *If A is a divisor of F/K and $deg\ A \geq 2g - 1$, then*

$$dim\ A = deg\ A + 1 - g.$$

Proof. Let W be a canonical divisor. Then

$$\dim A = \deg\ A + 1 - g + \dim(W - A)$$

by Theorem 2.2. Now $deg\ A \geq 2g - 1$ and $deg\ W = 2g - 2$, so

$$deg(W - A) < 0.$$

Now since

$$deg(W - A) < 0,$$

then

$$dim(W - A) = 0$$

from Lemma 2.41. Therefore,

$$dim\ A = deg\ A + 1 - g.$$

□

Proposition 2.44. *Let $P \in \mathbb{P}_F$. Then for any $n \geq 2g$, there exists a function $f \in F$ such that $(f)_\infty = nP$.*

Proof. We have

$$dim((n-1)P) = (n-1)deg\ P + 1 - g$$

and

$$dim(nP) = n \cdot deg\ P + 1 - g$$

from Corollary 2.43. Thus $\mathcal{L}((n-1)P) \subsetneqq \mathcal{L}(nP)$. Therefore, there exists $f \in \mathcal{L}(nP)$ such that $f \notin \mathcal{L}((n-1)P)$. Thus $(f)_\infty = nP$. □

Let $\mathbb{Z}^+$ denote the set of positive integers and $\mathbb{N}$ denote the set of nonnegative integers.

Definition 2.45. *Let $P \in \mathbb{P}_F$. An integer $n \geq 0$ is called a pole number of P if and only if there is a function $f \in F$ such that $(f)_\infty = nP$. If $n \geq 0$ is not a pole number, then n is called a gap number.*

We define the set of integers n such that n is not a gap number by

$$H(P) := \{n \in \mathbb{N} : \exists f \in F \text{ such that } (f)_\infty = nP\}.$$

The set of all integers n such that n is a gap number is the gap set

$$G(P) := \{n \in \mathbb{N} : (f)_\infty \neq nP \; \forall \; f \in F\} = \{n \in \mathbb{N} : n \notin H(P)\}.$$

Proposition 2.46. *The set $H(P)$ is an additive semigroup.*

Proof. Suppose that $n_1, n_2 \in H(P)$. Then there exist functions f_1 and f_2 in F such that

$$(f_1)_\infty = n_1 P$$

and

$$(f_2)_\infty = n_2 P.$$

Now

$$(f_1 f_2)_\infty = (n_1 + n_2)P.$$

Thus $n_1 + n_2 \in H(P)$. □

Recall that each principal divisor has degree zero. The next result shows that not every divisor of degree zero is a principal divisor.

Theorem 2.3. *(Weierstrass Gap Theorem). Suppose that F/K has genus $g > 0$ and P is a place of degree one. Then there are exactly g gap numbers $i_1 < \cdots < i_g$ of P and*

$$i_1 = 1 \textit{ and } i_g \leq 2g - 1.$$

Proof. If $n > 2g - 1$, then n is not a gap number from Proposition 2.44. Also, 0 is not a gap number because $(1) = 0$. We can characterize gap numbers in the following manner:

$$i \text{ is a gap number of P if and only if } \mathcal{L}((i-1)P) = \mathcal{L}(iP).$$

By Lemma 2.38, we have the following containment of vector spaces

$$K = \mathcal{L}(0) \subseteq \mathcal{L}(P) \subseteq \mathcal{L}(2P) \subseteq \cdots \subseteq \mathcal{L}((2g-1)P).$$

Note that $dim\ \mathcal{L}(0) = 1$ and $dim\ \mathcal{L}((2g-1)P) = deg\ A + 1 - g = 2g - 1 + 1 - g = g$ by Corollary 2.43. By Lemma 2.38,

$$dim\ \mathcal{L}(iP) \leq dim\ \mathcal{L}((i-1)P) + 1.$$

for any i. So there are exactly $g-1$ numbers $i, 1 \leq i \leq 2g-1$ such that

$$\mathcal{L}((i-1)P) \subsetneqq \mathcal{L}(iP).$$

This leaves g remaining integers that are gap numbers. Finally, we must show that 1 is a gap number. Suppose for the sake of contradiction that 1 is not a gap number. Then, since pole numbers form an additive semigroup, all $n \in \mathbb{Z}^+$ are pole numbers and there are no gap numbers. This contradicts that $|G(P)| = g > 0$. Therefore, 1 is a gap number. □

Example 2.47. *Suppose F/K has genus $g > 0$ and $P, Q \in \mathbb{P}_F$ are places of degree one. Then $P - Q$ is a divisor of degree zero, but $P - Q$ is not a principal divisor.*

Given $a_1, \ldots, a_k \in \mathbb{Z}^+$,

$$\langle a_1, \ldots, a_k \rangle := \left\{ \sum_{i=1}^{k} c_i a_i : c_i \in \mathbb{N} \right\}$$

is the semigroup generated by $a_1, \ldots, a_k$.

Example 2.48. *Consider the Hermitian function field $\mathbb{F}_{q^2}(x,y)/\mathbb{F}_{q^2}$ where $y^q + y = x^{q+1}$. We claim that $H(P_\infty) = \langle q, q+1 \rangle$.*

Since

$$(x) = \sum_{b^q + b = 0} P_{0b} - qP_\infty,$$

$q \in H(P_\infty)$. *Because*

$$(y) = (q+1)P_{00} - (q+1)P_\infty,$$

$q+1 \in H(P_\infty)$. *Since $H(P_\infty)$ is a semigroup under addition,*

$$\langle q, q+1 \rangle \subseteq H(P_\infty).$$

Next it must be shown that q and $q+1$ generate $H(P_\infty)$. By Theorem 2.3,

$$|G(P_\infty)| = g = \frac{q(q-1)}{2}.$$

Notice that

$$\begin{aligned} |\mathbb{N} \setminus \langle q, q+1 \rangle\ | &= \tfrac{q(q+1)-q-(q+1)+1}{2} \\ &= \tfrac{q^2+q-q-q-1+1}{2} \\ &= \tfrac{q(q-1)}{2} \\ &= |G(P_\infty)|. \end{aligned}$$

Thus, $H(P_\infty) = \langle q, q+1 \rangle$.

We can associate a projective curve with a function field F/K. The following discussion, as well as [33, App. B], details this.

Define a relation $\sim$ on $K^3 \setminus \{(0,0,0)\}$ by

$$(a,b,c) \sim (a',b',c')$$

if and only if there exists $\lambda \in K \setminus \{0\}$ such that $a = \lambda a',\ b = \lambda b'$, and $c = \lambda c'$.

Proposition 2.49. *The relation $\sim$ is an equivalence relation on $K^3 \setminus \{(0,0,0)\}$.*

Proof. In order to show that $\sim$ is an equivalence relation, we must show that $\sim$ is reflexive, symmetric, and transitive.

We first show that $\sim$ is reflexive. Let $(a,b,c) \in K^3 \setminus \{(0,0,0)\}$. Then $\lambda = 1 \in K \setminus \{0\}$ since K is a field. Thus we have $(a,b,c) \sim (a,b,c)$.

We now show that $\sim$ is symmetric. Let $(a,b,c),(a',b',c') \in K^3 \setminus \{(0,0,0)\}$ such that $(a,b,c) \sim (a',b',c')$. Then, for some $\lambda \in K \setminus \{0\}$, we have $a = \lambda a',\ b = \lambda b'$, and $c = \lambda c'$. Now since K is a field, then $\lambda \in K \setminus \{0\} \Rightarrow \lambda^{-1} \in K \setminus \{0\}$. Thus we have

$$a' = \lambda^{-1}a,\ b' = \lambda^{-1}b, \text{ and } c' = \lambda^{-1}c.$$

Therefore, $(a', b', c') \sim (a, b, c)$.

Finally we show that $\sim$ is transitive. Let $(a, b, c), (a', b', c'), (a'', b'', c'') \in K^3 \setminus \{(0, 0, 0)\}$ such that $(a, b, c) \sim (a', b', c')$ and $(a', b', c') \sim (a'', b'', c'')$. Then we have

$$a = \lambda a',\ b = \lambda b',\ c = \lambda c' \text{ and } a' = \lambda' a'',\ b' = \lambda' b'',\ c' = \lambda' c''$$

for some $\lambda, \lambda' \in K \setminus \{0\}$. Thus we have the following set of equalities:

$$a = \lambda a' = \lambda\lambda' a'',\ b = \lambda b' = \lambda\lambda' b'',\ c = \lambda c' = \lambda\lambda' c''.$$

We can see that $\lambda\lambda' \in K \setminus \{0\}$, because K is a field. Therefore

$$(a, b, c) \sim (a'', b'', c'').$$

Therefore, $\sim$ is an equivalence relation. □

Given $a, b, c \in K$ where a, b, and c are not all zero, let $(a : b : c)$ denote the equivalence class of (a, b, c) under $\sim$; hence,

$$(a : b : c) = \{(\lambda a, \lambda b, \lambda c) :\ \lambda \in K \setminus \{0\}\}.$$

Definition 2.50. *Let K be a field. The projective plane over K is defined by*

$$\mathbb{P}^2_K := \left\{(a : b : c) : (a, b, c) \in K^3 \setminus \{(0, 0, 0)\}\right\}.$$

The equivalence classes $(a : b : c)$ are called points in the projective plane.

If $(a : b : c) \in \mathbb{P}^2_K$ and $c \neq 0$, then

$$(a : b : c) = (c^{-1}a : c^{-1}b : 1).$$

If $(a : b : 0) \in \mathbb{P}^2_K$ and $b \neq 0$, then

$$(a : b : 0) = (b^{-1}a : 1 : 0).$$

If $(a:0:0) \in \mathbb{P}_K^2$, then $a \neq 0$ and

$$(a:0:0) = (1:0:0).$$

Therefore,

$$\mathbb{P}_K^2 = \{(a:b:1) : a, b \in K\} \cup \{(a:1:0) : a \in K\} \cup \{(1:0:0)\}.$$

The points $(a:b:1)$ are called affine points, while those of the form $(a:b:0)$ are called points at infinity. If $f(x,y) \in K[x,y]$, then the associated homogeneous polynomial $F[X,Y,Z] \in K[X,Y,Z]$ is given by

$$F[X,Y,Z] := Z^d f\left(\frac{X}{Z}, \frac{Y}{Z}\right) \in K[X,Y,Z],$$

where d denotes the degree of f.

Example 2.51. *Let q be a power of some prime number. If*

$$f(x,y) = y^q + y - x^{q+1} \in \mathbb{F}_{q^2}[x,y],$$

then deg$(f) = q+1$, and the associated homogeneous polynomial is

$$F(X,Y,Z) = ZY^q + Z^qY - X^{q+1} \in \mathbb{F}_{q^2}[X,Y,Z].$$

Definition 2.52. *Let K be a field and let $f(x,y) \in K[x,y]$ be a polynomial of degree d. The projective curve X defined by f is*

$$X = \left\{(a:b:c) \in \mathbb{P}_{\bar{K}}^2 : F(a,b,c) = 0\right\}$$

where $\bar{K}$ denotes the algebraic closure of K and F is the homogeneous polynomial associated with f.

We say that $(a : b : c)$ is a point on X provided $(a : b : c) \in X$. If $a, b, c \in K$, then we say that $(a : b : c)$ is a K-rational point on X. If $c = 1$ (or $(a : b : c) \sim (a' : b' : 1)$) for some a', b', then $(a : b : c :)$ is called an affine point; otherwise (a:b:c) is called a point at infinity.

Example 2.53. *Consider the Hermitian curve defined by $y^q + y = x^{q+1}$. The associated homogeneous equation is $y^q z + yz^q = x^{q+1}$. Thus the points on the Hermitian curve in the projective plane over $\mathbb{F}_{q^2}$ are $(a : b : c) \in \mathbb{P}^2_{\mathbb{F}_{q^2}}$ where*

$$b^q c + bc^q = a^{q+1}.$$

If $c = 0$, then we have the point $(a : b : 0)$ where $b^q(0) + b(0)^q = a^{q+1}$. Hence, $a = 0$. Thus $(a : b : 0) = (0 : b : 0) = (0 : 1 : 0)$ is the unique point at infinity on the Hermitian curve. The affine points are $(a : b : 1)$ where $b^q + b = a^{q+1}$.

Looking at Example 2.20, we see that there is a one-to-one correspondence between these points on the projective plane curve and the places of degree one of the corresponding function field.

Definition 2.54. *Let K be a field and $f(x, y) \in K[x, y]$ and let F be the associated homogeneous polynomial. Let $F_x(x, y, z)$, $F_y(x, y, z)$, and $F_Z(x, y, z)$ be the partial derivatives of F with respect to x, y, and z respectively. A point $(x_0, y_0, z_0) \in \mathbb{F}^2$ on a curve X is singular if*

1. *$F(x_0, y_0, z_0) = 0$,*
2. *$F_x(x_0, y_0, z_0) = 0$,*
3. *$F_y(x_0, y_0, z_0) = 0$, and*
4. *$F_z(x_0, y_0, z_0) = 0$.*

A curve is said to be nonsingular if it has no singular points.

For further discussion of curves, see [36].

The notions of divisor, divisor group, and Riemann-Roch space as defined for function fields earlier are valid for curves, with place replace by point.

2.3 Algebraic Geometry Codes

We now discuss algebraic geometry codes. These codes have attracted much attention because they give rise to a sequence of codes with parameters that exceed the Gilbert-Varshamov bound. Moreover, they can be viewed as generalizations of Reed-Solomon codes.

Definition 2.55. *Let X be a nonsingular projective curve over $\mathbb{F}_q$. Let G and $D := P_1 + \cdots + P_n$ be divisors on X such that $supp(D) \cap supp(G) = \emptyset$ and D is supported by n distinct $\mathbb{F}_q$-rational points. The algebraic geometry code $C_{\mathcal{L}}(D, G)$ defined by G and D is*

$$C_{\mathcal{L}}(D, G) := \{(f(P_1), ..., f(P_n)) : f \in \mathcal{L}(G)\} \subseteq \mathbb{F}_q^n.$$

We sometimes refer to an algebraic geometry code as an AG code. Alternatively, given a function field $F/\mathbb{F}_q$ and divisors $G, D \in \mathcal{D}_F$, one may define an AG code as above.

Definition 2.56. *If $C_{\mathcal{L}}(D, G)$ is an algebraic geometry code such that G is a linear combination of m distinct places (or points), then we call $C_{\mathcal{L}}(D, G)$ an m-point code. If $m > 1$, we say that $C_{\mathcal{L}}(D, G)$ is a multipoint code.*

Example 2.57. *Consider a function field $F/\mathbb{F}_q$. Let $G = mP$ where m is a positive integer and $P \in \mathbb{P}_F$ and let D be the sum of the places of degree one not contained in the support of G. Then $C_{\mathcal{L}}(D, G)$ is a one-point code.*
Let $G' = aP_1 + bP_2$ for some nonzero integers a and b and $P_1, P_2 \in \mathbb{P}_F$, and let D be the sum of the places of degree one not contained in the support of G'. Then $C_{\mathcal{L}}(D, G')$ is a two-point code.

Example 2.58. *Consider a $[q-1, k, q-k]$ Reed-Solomon code C over $\mathbb{F}_q$ as described in Example 2.10. To view the Reed-Solomon code C as a one-point AG code, let $X = \mathbb{P}^2_{\mathbb{F}_q}$ and $\alpha_1, \ldots, \alpha_{q-1} = \mathbb{F}_q \backslash \{0\}$. Consider the points $P_{\alpha_1} := (\alpha_1 : 1), \ldots, P_{\alpha_{q-1}} := (\alpha_{q-1} : 1)$, $P_\infty := (1 : 0)$ on the projective line $\mathbb{P}^1_{\mathbb{F}_q}$. Then*

$$C = \{f(P_{\alpha_1}), \ldots, f(P_{\alpha_{q-1}}) | f \in \mathcal{L}((k-1)P_\infty)\},$$

illustrating that a Reed-Solomon code is a one-point AG code.

Example 2.59. *Consider the Hermitian curve $y^q + y = x^{q+1}$ over $\mathbb{F}_{q^2}$. Let*

$$G = \sum_{b^q+b=0} a_b P_{0b}$$

for some nonzero integers a_b. Set D to be the sum of the places not contained in the support of G. Then $C_{\mathcal{L}}(D, G)$ is a q-point code of length $q^3 + 1 - q$.

Theorem 2.4. *Suppose that G and D are as defined in Definition 2.55. Then $C_{\mathcal{L}}(D, G)$ is an $[n, k, d]$ code with parameters*

$$k = dim\ G - dim(G - D)$$

and

$$d \geq n - deg\ G.$$

Proof. Define an evaluation map

$$\begin{array}{rccc} ev: & \mathcal{L}(G) & \longrightarrow & \mathbb{F}_q^n \\ & f & \mapsto & (f(P_1), \ldots, f(P_n)). \end{array}$$

Then ev is a surjective linear map from $\mathcal{L}(G)$ to $C_{\mathcal{L}}(D, G)$ and has kernel

$$Ker(ev) = \{f \in \mathcal{L}(G) : v_{P_i}(x) > 0 \text{ for } i = 1, \ldots, n\} = \mathcal{L}(G - D).$$

Thus

$$k = \dim C_{\mathcal{L}}(D, G) = \dim \mathcal{L}(G) - \dim \mathcal{L}(G - D) = \dim G - \dim(G - D).$$

Now consider the minimum distance d of $C_{\mathcal{L}}(D, G)$. Assume $C_{\mathcal{L}}(D, G) \neq \{(0, \ldots, 0)\}$. Choose $f \in \mathcal{L}(G)$ with $wt(ev(f)) = d$. Then there are exactly $n - d$ places $P_{i_1}, \ldots, P_{i_{n-d}}$ in the support of D that are zeros of f. Thus,

$$f \in \mathcal{L}(G - (P_{i_1} + \cdots + P_{i_{n-d}})) \setminus \{0\}.$$

Then, by Lemma 2.41, $0 \leq deg(G - (P_{i_1} + \cdots + P_{i_{n-d}})) = deg\ G - n + d$. Therefore $d \geq n - deg\ G$. □

Corollary 2.60. *Suppose that the degree of G is strictly less than n. Then the evaluation map $ev : \mathcal{L}(G) \to C_{\mathcal{L}}(D, G)$ is injective, and $C_{\mathcal{L}}(D, G)$ is an $[n, k, d]$ code with*

$$k = dim\ G + 1 - g$$

and

$$d \geq n - deg\ G.$$

Thus,

$$k + d \geq n + 1 - g.$$

Proof. Suppose that the degree of G is strictly less than n. Then

$$deg(G - D) = deg\ G - n < 0$$

and

$$\mathcal{L}(G - D) = \{0\}$$

by Lemma 2.41.

The kernel of the evaluation map ev as found in the proof of Theorem 2.4 is

$$\mathcal{L}(G - D) = \{0\};$$

hence, ev is injective. Then by combining the fact that $d \geq n - deg\ G$ along with the Riemann-Roch Theorem we have

$$d \geq n - deg\ G$$

and

$$k = dim\ G + 1 - g.$$

Thus,

$$k + d \geq n + 1 - g.$$

□

By the Singleton Bound, an $[n, k, d]$ code satisfies

$$k + d \leq n + 1.$$

Notice that given an $[n, k, d]$ AG code $C_{\mathcal{L}}(D, G)$ with $\deg G < n$,

$$n + 1 - g \leq k + d \leq n + 1.$$

Definition 2.61. *Let C be an $[n, k]$ code over $\mathbb{F}_q$. A generator matrix for C is any $k \times n$ matrix whose rows form a basis for C as an $\mathbb{F}_q$ vector space. A parity check matrix for C is any generator matrix for $C^{\perp}$.*

Definition 2.62. *Suppose that G and D are as in Definition* 2.55. *Then*

$$C_{\Omega}(D, G) := C_{\mathcal{L}}(D, G)^{\perp}.$$

Fact 2.63. *The dual of an AG code is an AG code. Specifically,*

$$C_{\mathcal{L}}(D,G)^{\perp} = C_{\Omega}(D,G) = C_{\mathcal{L}}(D,H)$$

with $H := D - G + W$ *for some canonical divisor* W *such that* $v_{P_i}(W) = 1$ *for all* P_i *in the support of* D.

Proof. See [33, Proposition II.2.10]. □

Example 2.64. *Consider a Hermitian code* $C_{\mathcal{L}}(D, mP_\infty)$ *where*

$$0 \leq m \leq q^3 + q^2 - q - 2$$

and $D = \sum_{i=1}^{q^3} P_i$ *with* $P_i \neq P_\infty$ *for all* i. *The dual of this code is*

$$C_{\mathcal{L}}(D, mP_\infty)^{\perp} = C_{\mathcal{L}}(D, (q^3 + q^2 - q - 2 - m)P_\infty).$$

To see this take $W = (q^3 + q^2 - q - 2)P_\infty - D$ *in Fact 2.63. Therefore, the dual of a one-point Hermitian code supported by* P_∞ *is also a one-point Hermitian code.*

Example 2.65. *Consider a Hermitian code* $C_{\mathcal{L}}(D, aP_\infty + bP_{00})$ *where*

$$0 \leq a, b \leq q^3 + q^2 - q - 2$$

and $D = \sum_{i=1}^{q^3-1} P_i$ *with* $P_i \neq P_\infty$ *and* $P_i \neq P_{00}$ *for all* i. *The dual of this code is*

$$C_{\mathcal{L}}(D, aP_\infty + bP_{00})^{\perp} = C_{\mathcal{L}}(D, (q^3 + q^2 - q - 2 - a)P_\infty - (b+1)P_{00}).$$

To see this take $W = (q^3 + q^2 - q - 2)P_\infty - P_{00} - D$ *in Fact 2.63. Hence, the dual of a two-point Hermitian code supported by* P_{00} *and* P_∞ *is also a two-point Hermitian code supported by* P_{00} *and* P_∞.

CHAPTER 3
LIST DECODING

We now shift our attention to decoding. A standard decorder returns the unique codeword closest to a received word, if such a word exists. A list decoder, on the other hand, returns a list of codewords within a specified distance of a received word. Specifically, a list decoder returns all codewords within a particular Hamming sphere about a received word.

In this chapter, we provide an overview of the seminal papers on list decoding Reed-Solomon codes and AG codes. We first explore Sudan's original list decoding algorithm for Reed-Solomon codes [34] and the later refinement by Guruswami and Sudan [15]. Next, we consider the work of Shokrollahi and Wasserman on list decoding AG codes [32] as well as the Guruswami-Sudan algorithm for one-point codes [15]. Then we discuss different parameter choices for the Guruswami-Sudan algorithm for one-point codes that give a better bound on the list size. We continue by showing an analagous improvement for the list decoding algorithm for AG codes defined over rings presented by [5]. We close the chapter with a discussion of list decoding for one-point codes supported by places of higher degree.

3.1 List Decoding Reed-Solomon Codes

Although list decoding was first introduced in 1957 and 1958 by Elias [11] and Wozencraft [39] respectively, it went largely unnoticed until the 1990's. This was due in part to the lack of efficient list decoding algorithms. In the early 1990's, however, Sudan [34] developed an algorithm for decoding Reed-Solomon codes that ran in polynomial time. The Sudan algorithm requires the Reed-Solomon code to be low rate. Refinements to the Sudan algorithm were made by Guruswami and Sudan [15] in order to remove this restriction. In this section, we present the main ideas of the Sudan algorithm and the Guruswami-Sudan algorithm.

Both the Sudan algorithm and the Guruswami-Sudan algorithm use the notion of (a, b)-weighted degree as defined next.

Definition 3.1. *Let $Q(x, y) \in \mathbb{F}[x, y]$. Given $a, b \in \mathbb{Z}^+$, the (a, b)-weighted degree of a polynomial $Q(x, y) = \sum_i \sum_j c_{ij} x^i y^j$ is given by $\max_{i,j}\{ia + bj\}$.*

3.1.1 Sudan List Decoding Algorithm for Reed-Solomon Codes

The Sudan algorithm has three main steps: initialization, interpolation, and factorization (or root-finding). One of the parameters that is set in the initialization step is an agreement parameter t. This parameter determines the number of coordinates in which a codeword must agree with a received word in order for the associated function to be included on the output list.

Algorithm 3.2. *(Sudan list decoding algorithm for Reed-Solomon codes)*
Let C be an $[q - 1, k, q - k]$ Reed-Solomon code defined over $\mathbb{F}_q = \{\alpha_1, \ldots, \alpha_q\}$ as defined in Example 2.10.
The input consists of a received word $w \in \mathbb{F}_q^{q-1}$, a parameter s related to the degree of the polynomial to be found, and an agreement parameter t.

0. *Initialize: Fix parameters m and l, both positive integers.*
1. *Interpolate: Find $Q(x, y) \in \mathbb{F}_q[x, y] \setminus \{0\}$ of degree at most s such that*
   (a) *$Q(x, y)$ has $(1, d)$-weighted degree at most $m + ld$ and*
   (b) *$Q(\alpha_i, w_i) = 0$ for all i with $f(\alpha_i) = w_i$.*
2. *Factor: Determine the roots of Q in order to output a list of functions h such that $d(ev(h), w) \leq n - t$. When we say root here, we mean a function $h(x)$ such that $y - h(x)$ is a factor of Q.*

Consider the $[q-1, k, q-k]$ Reed-Solomon code C which is obtained by evaluating polynomials of degree less than k at each of the elements in $\mathbb{F}_q$ where $\alpha_1, \ldots, \alpha_n$ are elements of $\mathbb{F}_q$. Let $t < n \in \mathbb{Z}^+$ be an agreement parameter and suppose that $(w_1, \ldots, w_n) \in \mathbb{F}_q^n$ is the received word. The goal of Sudan's algorithm is to output a list, $\Omega = \{h_1, \ldots, h_j\}$ such that there is a codeword $y = ev(h_i) \in C$ with $d(ev(h_i), w) \leq n - t$ if and only if $h_i \in \Omega$.

In the initialization step, the parameters m and l are set so that the interpolating polynomial Q exists and whenever there is a codeword $y = ev(h)$ with $d(ev(h_i), w) \leq n-t$, the interpolating polynomial Q has $h(x)$ as a root.

In the interpolation step, an interpolating polynomial $Q(x, y)$ is found that satisfies the following three restrictions:

1. the $(1, k)$-weighted degree of $Q(x, y)$ is bounded by $m + ld$,
2. $Q(\alpha_i, w_i) = 0$ for $0 \leq i \leq n$, and
3. $Q(x, y)$ is not identically 0.

The interpolating polynomial has the form

$$Q(x, y) = \sum_{j_1=0}^{l} \sum_{j_2=0}^{m+(l-j_1)k} q_{j_2 j_1} x^{j_2} y^{j_1} \in \mathbb{F}_q[x, y].$$

For this polynomial, the $q_{j_2 j_1}$'s are unknowns and for each pair (α_i, w_i) the constraint

$$0 = \sum_{j_1=0}^{l} \sum_{j_2=0}^{m+(l-j_1)k} q_{j_2 j_1} \alpha_i^{j_2} w_i^{j_1}$$

must be satisfied. To ensure the existence of such a non-zero polynomial, Sudan required the number of unknowns be greater than the number of constraints.

In the factorization step, all functions f such that $\deg(f) < k$ and $f(\alpha_i) = w_i$ for at least t values of i are found to be factors of $Q(x, y)$. Sudan accomplishes this by requiring the degree of $Q(x, f(x))$ be less than the number of zeros of $Q(x, f(x))$, so that

$Q(x, f(x))$ is identically 0 for all f such that $\deg(f) < k$ and $f(\alpha_i) = w_i$ for at least t values of i .

The restrictions in the interpolation and factorization steps are

$$t > m + lk$$

and

$$(m+1)(l+1) + k\binom{l+1}{2} > n.$$

To satisfy these constraints, Sudan set

$$m = \left\lceil \frac{k}{2} \right\rceil - 1$$

and

$$l = \left\lceil \sqrt{\frac{2(n+1)}{k}} \right\rceil - 1$$

as choices for m and l in the initialization step.

3.1.2 Guruswami-Sudan List Decoding Algorithm for Reed-Solomon Codes

Although Sudan's algorithm ran in polynomial time, it worked only for codes of low rate. It is desirable to have an algorithm without this restriction as higher rate codes are better suited to some applications. In [15], the authors accomplish this by requiring more from the interpolating polynomial than Sudan had in [34]. Specifically, they define a singularity and require that the interpolating polynomial have a singularity of degree larger than one at each of the points; in this terminology, Sudan had required a singularity of degree one. We will make this idea more precise by presenting the Guruswami-Sudan algorithm here.

As in Sudan's algorithm, the Guruswami-Sudan algorithm has three main steps.

Algorithm 3.3. *(Guruswami-Sudan list decoding algorithm for Reed-Solomon codes) Let C be a $[q-1, k, q-k]$ Reed-Solomon code defined over $\mathbb{F}_q$ as defined in Example 2.10.*

The input consists of a received word $w \in \mathbb{F}_q^{q-1}$, the code dimension k, and agreement parameter t.

0. *Initialize: Fix parameters r and l, both positive integers.*
1. *Interpolate: Find $Q(x, y) \in \mathbb{F}_q[x, y] \setminus \{0\}$ such that*
   (a) *the $(1, k)$-weighted degree of Q is at most l, and*
   (b) *$f(x)$ is a root of Q for all f such that $d(ev(f), w) \leq n - t$.*
2. *Factor: Determine the roots $h_i \in \mathbb{F}_q[x]$ of $Q(x, y)$ in order to output a list of functions of degree at most k such that $d(ev(h), w) \leq n - t$.*

Consider the $[q-1, k, q-k]$ Reed-Solomon code C which is obtained by evaluating polynomials of degree less than k at each of the elements in $\mathbb{F}_q$ with $\alpha_1, \ldots, \alpha_n$ the elements of $\mathbb{F}_q$. Let $t < n \in \mathbb{Z}^+$ be an agreement parameter and suppose that $(w_1, \ldots, w_n) \in \mathbb{F}_q^n$ is a received word. The goal of the Guruswami-Sudan algorithm is to output a list of functions, $\Omega = \{h_1, \ldots, h_j\}$ such that there is a codeword $y = ev(h_i) \in C$ with $d(ev(h_i), w) \leq n - t$ if and only if $h_i \in \Omega$.

In Sudan's algorithm, the requirement was that for every pair (α_i, w_i) and function f such that $f(\alpha_i) = w_i$,

$$(x - \alpha_i) | Q(x, f(x)).$$

The additional requirement that Guruswami and Sudan enforce is that not only does $(x - \alpha_i)$ divide $Q(x, f(x))$ but also

$$(x - \alpha_i)^r | Q(x, f(x))$$

where r is a predetermined parameter. This additional requirement, that we have a singularity of degree r, increases the number of unknowns for the interpolating polynomial significantly more that than it increases the number of constraints. Again, as in Sudan's algorithm, the requirement that the number of unknowns is greater than the number of constraints. This restriction manifests itself in the following inequality

$$n\binom{r+1}{2} < \frac{(rt-1)(rt+1)}{2k}$$

and the additional restriction $rt > l$ also is enforced.

Based on these restrictions, Guruswami and Sudan set

$$l := rt - 1$$

and

$$r := \left\lfloor \frac{kn + \sqrt{k^2n^2 + 4(t^2 - kn)}}{2(t^2 - kn)} \right\rfloor + 1.$$

The choice of l is the "obvious" choice satisfying the constraint $rt > l$. The choice of r, however, is obtained by reformulating the inequality

$$n\binom{r+1}{2} < \frac{(rt-1)(rt+1)}{2k}$$

as a quadratic and choosing r to be just greater than the floor of the larger of the two roots. In [38], Wang provides other choices for r and s which result in a lower degree interpolating polynomial.

3.2 List Decoding AG Codes

The renewed interest in list decoding brought about by Sudan's algorithm was not restricted to Reed-Solomon codes. Namely, it sparked interest in list decoding algebraic geometry codes which, as illustrated in Example 2.58, are a generalization of Reed-Solomon codes. Sudan's algorithm was adapted by Shokrollahi and Wasserman in 1998 [32] to handle low-rate AG codes. The later work by Guruswami and Sudan [15] eliminated the low-rate restriction not only for Reed-Solomon codes, but also for one-point algebraic geometry codes. In this section, we will discuss both of these algorithms.

3.2.1 Shokrollahi-Wasserman List Decoding Algorithm for AG Codes

As in the list decoding algorithms for Reed-Solomon codes, the list decoding algorithms for AG codes consist of three main steps: initialization, interpolation, and factorization.

Consider the algebraic geometry code $C_{\mathcal{L}}(D, G)$ defined using a curve X of genus g. Suppose that a basis for $C_{\mathcal{L}}(D, G)$ is given. Let H be a divisor such that the support of H does not intersect the support of D and

$$\deg(H) = t - 1 - b \cdot \deg(G) = \left\lceil \frac{n+1}{b+1} - \frac{b \cdot \deg(G)}{2} + g - 1 \right\rceil .$$

The divisor H will be used in determining a basis for the Riemann-Roch space and the degree will prove to to be what is needed to ensure that the interpolating polynomial exists.

Suppose that w is a received word. The goal is to recover all codewords $y = (y_1, \ldots, y_n)$ that agree with w in at least t places.

In the interpolation step, an interpolating polynomial

$$Q(T) = \mu_b T^b + \cdots + \mu_1 T + \mu_0 \in K[T]$$

is found where K denotes the function field associated with X. For convenience we set

$$Q(P_j, w_j) := \sum_{i=0}^{b} \mu_i(P_j) w_j^i.$$

Then $Q(T)$ should satisfy the following three restrictions:

1. $\mu_i \in C_{\mathcal{L}}(H + (b - i)G)$ for all $0 \leq i \leq b$ and
2. $Q(P_j, w_j) = 0$ for $0 \leq j \leq n$, and
3. $Q(T) \neq 0$.

As in the interpolation step in both Sudan's algorithm and the Guruswami-Sudan algorithm for Reed-Solomon codes, finding such a polynomial is based on solving a system of linear equations. Namely, the restriction that $Q(P_j, w_j) = 0$ can be written more precisely as

$$\sum_{i=0}^{b} \mu_i(P_j)w_j^i = 0 \text{ for all } 1 \leq j \leq n.$$

The number of unknowns is given by $\sum_{0\leq i\leq b} dim(H+iG)$. Since

$$dim(H+iG) \geq deg(H) + i \cdot \deg(G) - g + 1$$

by the Riemann-Roch Theorem, the number of unknowns is greater than the number of equations. Thus, a nonzero solution to this system exists.

In the factorization step, the roots $h \in K$ of $Q(T)$ are found. In particular, every function h such that

$$d(ev(h), ev(f)) \leq n - t - 2$$

are found; the authors show that every such function is indeed seen as a root of $Q(T)$. This is attained by considering $Q(h)$ and finding that its number of zeros is not equal to its number of poles. Thus, $Q(h) = 0$; that is, h is a root of $Q(T)$.

Therefore, this decoding algorithm returns all codewords y that agree with a received word w in at least t positions.

Definition 3.4. *Let $e, b \in \mathbb{Z}^+$. A code of length n over $\mathbb{F}_q$ is said to be (e, b)-decodable if every Hamming-Sphere of radius e in $\mathbb{F}_q^n$ contains at most b codewords.*

Thus, the Shokrollahi-Wasserman algorithm illustrates that an AG code $C_{\mathcal{L}}(D, G)$ over $\mathbb{F}_q$ with $\deg G = \alpha$ is $(n-t-2, b)$-decodable for any $b \in \mathbb{Z}^+$ with

$$t := \left\lceil \frac{n+1}{b+1} + \frac{b\alpha}{2} + g - 1 \right\rceil + 1$$

and

$$\alpha := k + g - 1.$$

3.2.2 Guruswami-Sudan List Decoding Algorithm for One-point Codes

Although the Shokrollahi-Wasserman algorithm applies to multipoint AG codes, it works only for AG codes of low rate. We would like to remove this restriction. In [15], the authors do just that for a restricted class of AG codes. In particular, they do this for one-point codes. They accomplish this by requiring more from the interpolating polynomial than Shokrollahi and Wasserman had in [32]. Specifically, they require that the interpolating polynomial have a singularity of degree larger than one. We present this algorithm here.

As in the previous algorithms, there are the three steps: initialization, interpolation, and factorization.

Algorithm 3.5. *(Guruswami-Sudan list decoding algorithm for one-point codes) Let $C := C_{\mathcal{L}}(D, \alpha P)$ be an $[n, k, d]$ code defined using a curve X of genus g over $\mathbb{F}_q$, and let K be the function field associated with X.*

The input consists of a received word $w \in \mathbb{F}_q^n$, α, and an agreement parameter t with $t^2 > \alpha n$. Let $\Omega = \{f : d(ev(f_i), w) \leq n - t \text{ and } f_i \in \mathcal{L}(\alpha P)\}$.

0. *Initialization: Fix parameters r and s, both positive integers.*
1. *Interpolation: Find $Q(T) \in K[T] \setminus \{0\}$ of degree at most s such that*
   (a) $Q(f) \in \mathcal{L}((rt-1)P) \; \forall \; f \in \mathcal{L}(\alpha P)$ and
   (b) $Q(f)$ has an r-singularity at P_i for all i with $f(P_i) = w_i$.
2. *Factorization: Determine the roots of $Q(T)$ in order to output a list of at most s functions $h \in \mathcal{L}(\alpha P)$ such that $d(ev(h), w) \leq n - t$.*

The initialization step consists of setting some parameters and values will be determined later.

The goal of the interpolation step is to find a polynomial $Q(T) \in K[T]$ of degree s such that each $f \in \mathcal{L}(\alpha P)$ that agrees with w in at least t places is a root of $Q(T)$. Moreover, there is also a requirement that $Q(f) \in \mathcal{L}((rt-1)P)$ and $Q(f)$ has an r-singularity at P_i for all i with $f(P_i) = w_i$.

The polynomial Q looks like

$$Q(T) = \sum_{j=0}^{s} \sum_{i=0}^{l-g+1-\alpha j} q_{i\,j} \phi_i T^j$$

where

- $\phi_i \in \mathcal{L}((i+g-1)P)$ and
- ϕ_i all have *distinct* pole orders at P.

Moreover, there exist functions $\psi_1, \dots, \psi_k$ such that

$$\phi_i = \sum_{j=1}^{k} \alpha_{P,i,j} \psi_j \text{ with } \alpha_{P,i,j} \in \mathbb{F}_q.$$

Thus, Q can be expressed as

$$Q(P,T) = \sum_{j_2=0}^{s} \sum_{j_3=1}^{l-g+1} \sum_{j_1=1}^{l-g+1-\alpha j_2} q'_{j_1,j_2,j_3} \psi_{j_3,P}(P) T^{j_2}$$

where $Q(P,T) := Q(T)(P)$. If $q^{(i)}_{j_3,j_4} = 0$ for all $j_4 + j_3 \leq r$, then (P_i, w_i) is an r-singularity.

Next we discuss why the interpolating polynomial Q exists. This is guaranteed by Step (0) of the algorithm which forces the number of unknowns to be larger than the number of constraints. Note that the condition that we have an r-singularity enforces the constraints

$$q^{(i)}_{j_3,j_4} = 0 \quad \forall\, j_3 + j_4 \leq r,\ j_3 \geq 1,\ j_4 \geq 0.$$

So we have that the number of constraints is

$$\sum_{j_3=1}^{r} \sum_{j_4=0}^{r-j_3} 1 = \frac{r^2+r}{2}$$

for each of the n points, P_i. So, there are

$$n\binom{r+1}{2}$$

constraints. Additionally, the we have that the number of unknowns is

$$\sum_{j_2=1}^{\lfloor\frac{l-g}{\alpha}\rfloor}\sum_{j_1=0}^{l-g+1-\alpha j_2} 1 \geq \frac{(l-g)(l-g+2)}{2\alpha}.$$

Thus Guruswami and Sudan force the number of constraints to be less than the number of unknowns to ensure a nonzero polynomial Q exists. This is equivalent to choosing parameters r and s such that

$$n\binom{r+1}{2} < \frac{(l-g)(l-g+2)}{2\alpha}$$

in the initialization step.

Next, Guruswami and Sudan ensure that f is a factor of Q for each f such that $ev(f)$ agrees with the received word in at least t places. They note that if we require that $rt > l$, the function $Q(f)$ will have more zeros than poles. Hence, $Q(f) = 0$ which implies that f is a root of Q.

Thus, in the initialization step, Guruswami and Sudan choose r and s that meet the two restrictions noted above. The choices that they make are

$$r := \left\lfloor \frac{2gt + \alpha n + \sqrt{(2gt+\alpha n)^2 - 4\left(g^2-1\right)\left(t^2-\alpha n\right)}}{2\left(t^2-\alpha n\right)} \right\rfloor + 1$$

and

$$s := \left\lfloor \frac{l-g}{\alpha} \right\rfloor.$$

We will discuss better choices for these values in the next section.

Finally, in the factorization step, the polynomial Q is factored and among its roots are all functions f such that $ev(f)$ agrees with the received word in at least t coordinates.

Therefore, the complete version of their algorithm follows.

Algorithm 3.6 (Guruswami-Sudan Algorithm)**.** Input: n, α, $w \in \mathbb{F}_q^n$, t.

Assumptions: $t^2 > \alpha n$.

0. Parameter choices: Set
$$\begin{aligned} r &:= \left\lfloor \frac{2gt+\alpha n+\sqrt{(2gt+\alpha n)^2-4(g^2-1)(t^2-\alpha n)}}{2(t^2-\alpha n)} \right\rfloor + 1, \\ l &:= rt-1, \text{ and} \\ s &:= \left\lfloor \frac{l-g}{\alpha} \right\rfloor. \end{aligned}$$

1. Interpolation: Find a polynomial $Q(T) \in K[T] \setminus \{0\}$ of degree at most s satisfying the properties
   (a) $Q(f) \in \mathcal{L}((rt-1)P)$ and
   (b) $Q(f)$ has an r-singularity at P_i for all i with $f(P_i) = w_i$
   for all $f \in \Omega$.

2. Factorization: Find all roots $h \in \mathcal{L}(\alpha P)$ of the polynomial Q. For each such h, if $h(Q_i) = w_i$ for at least t values of i, then add h to the output list.

Output: $h_1, \ldots, h_s$ such that $d(w, ev(h_i)) \leq n - t$.

Guruswami and Sudan's use of an r-singularity actually allows for greater flexibility, in particular, it allows for the removal of the low-rate restriction. Note that the definition of r-singularity is predicated on a one-point code. Hence, this does not provide a list decoding algorithm for multipoint AG codes.

3.3 Improved Bounds on the List Size in the Guruswami-Sudan Algorithm for One-point Codes

The previous section outlines the decoding algorithm due to Guruswami and Sudan as found in [15]. In this section, we give improved parameter choices which can be used in Step (0) of Algorithm 3.6. Certainly, it is advantageous to choose the parameters that result in a smaller degree interpolating polynomial Q and yield a better bound s on the list size of the output. We show how to do this for any one-point AG code $C_{\mathcal{L}}(D, \alpha P)$

and agreement parameter $t > \sqrt{\alpha n}$ satisfying either

$$\alpha < 2g$$

or

$$t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right).$$

The restriction on t seems necessary to obtain a polynomial time algorithm; Guruswami and Rudra have evidence that a lower agreement parameter may lead to super-polynomially large lists as output [13].

Lemma 3.7. *Suppose n, α, g, and t satisfy*

1. $t^2 > \alpha n$, *and*
2. *either* $\alpha < 2g$ *or* $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha-2g} + \alpha - 2g\right)$.

Then the following statements are equivalent:

1. *There exist positive integers r and s such that*

$$(s+1)(rt-g) - \alpha\binom{s+1}{2} > n\binom{r+1}{2}.$$

2. *There exist positive integers r and s satisfying the following conditions:*

   (a) $r > \frac{\alpha(n-t)+2tg+\sqrt{\Delta_2}}{2(t^2-\alpha n)}$ *or* $r < \frac{\alpha(n-t)+2tg-\sqrt{\Delta_2}}{2(t^2-\alpha n)}$, *and*

   (b) $s_1 < s < s_2$,
   where

$$\begin{array}{rcl} s_1 & := & \frac{rt-\frac{\alpha}{2}-g-\sqrt{\Delta_1}}{\alpha}, \\ s_2 & := & \frac{rt-\frac{\alpha}{2}-g+\sqrt{\Delta_1}}{\alpha}, \\ \Delta_1 & := & (t^2-\alpha n)\,r^2 + (\alpha t - \alpha n - 2tg)\,r + \frac{\alpha^2}{4} + g^2 - \alpha g, \text{ and} \\ \Delta_2 & := & \alpha^2 n(n+\alpha-2t) + 4\alpha g n(t+g-\alpha). \end{array}$$

Proof. Assume n, α, g, and t satisfy

$$\text{(i) } t^2 > \alpha n \text{ and (ii) either } \alpha < 2g \text{ or } t < \tfrac{1}{2}\left(\tfrac{\alpha n}{\alpha-2g} + \alpha - 2g\right).$$

(1) ⇒ (2): Suppose there exist positive integers r and s such that

$$(s+1)\,(rt-g)-\alpha\binom{s+1}{2}>n\binom{r+1}{2}.$$

Then

$$\frac{\alpha}{2}s^2-(rt-g-\frac{\alpha}{2})s+\frac{r^2n+rn}{2}-rt+g<0.$$

Set

$$h_1(x):=\frac{\alpha}{2}x^2-(rt-g-\frac{\alpha}{2})x+\frac{r^2n+rn}{2}-rt+g.$$

Since $h_1(s)<0$ and $\frac{\alpha}{2}>0$, $h_1(x)$ must have two distinct real roots. Let Δ_1 denote the discriminant of $h_1(x)$. Then

$$\Delta_1=(t^2-\alpha n)r^2+(\alpha t-\alpha n-2tg)r+\frac{\alpha^2}{4}+g^2-\alpha g>0,$$

and the roots of $h_1(x)$ are

$$s_1:=\frac{rt-\frac{\alpha}{2}-g-\sqrt{\Delta_1}}{\alpha}$$

and

$$s_2:=\frac{rt-\frac{\alpha}{2}-g+\sqrt{\Delta_1}}{\alpha}.$$

Consequently, $h_1(s)=(s-s_1)(s-s_2)$ and $s_1<s<s_2$. Thus, (b) holds.

Next, we prove (a). To see this, set

$$h_2(x):=(t^2-\alpha n)x+(\alpha t-\alpha n-2tg)x+\frac{\alpha^2}{4}+g^2-\alpha g.$$

Then $h_2(r)=\Delta_1>0$. Let Δ_2 be the discriminant of $h_2(x)$. Then

$$\begin{aligned}\Delta_2 &= \alpha^2n(n+\alpha-2t)+4\alpha gn(t+g-\alpha)\\ &= \alpha n\left(\alpha n+\alpha^2+4g^2-4\alpha g-2t\left(\alpha-2g\right)\right).\end{aligned}$$

In the case $\alpha\le 2g$, we see that

$$\Delta_2>\alpha n\left(2\alpha^2+4g^2-4\alpha g-2t\left(\alpha-2g\right)\right)=\alpha n\left(2\left(t-\alpha\right)\left(2g-\alpha\right)+4g^2\right)\ge 0$$

since $\alpha < t$. Otherwise, $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)$. Here, we have

$$\Delta_2 > \alpha n\left(\alpha n + \alpha^2 + 4g^2 - 4\alpha g - (\alpha - 2g)\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)\right) = 0.$$

Then

$$h_2(r) = \left(r - \frac{\alpha(n-t) + 2tg + \sqrt{\Delta_2}}{2(t^2 - \alpha n)}\right)\left(r - \frac{\alpha(n-t) + 2tg - \sqrt{\Delta_2}}{2(t^2 - \alpha n)}\right)$$

which implies $r > \frac{\alpha(n-t)+2tg+\sqrt{\Delta_2}}{2(t^2-\alpha n)}$ or $r < \frac{\alpha(n-t)+2tg-\sqrt{\Delta_2}}{2(t^2-\alpha n)}$.

(2) ⇒ (1): Suppose there exist positive integers r and s satisfying (a) and (b). Taking $h_1(x)$ and Δ_1 as above, we see that the choice of r guarantees that $\Delta_1 \geq 0$ and the choice of s guarantees $h_1(s) < 0$. As a result,

$$(s+1)(rt - g) - \alpha\binom{s+1}{2} > n\binom{r+1}{2}.$$

□

Next, we indicate how Lemma 3.7 can be used in conjunction with Algorithm 3.6 to obtain a better bound on the list size.

Theorem 3.1. *Consider the AG code $C_{\mathcal{L}}(D, \alpha P)$ on a curve X of genus g over a finite field $\mathbb{F}$ where $D := Q_1 + \cdots + Q_n$. Suppose (i) $t^2 > \alpha n$ and (ii) either $\alpha < 2g$ or $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)$. Then taking*

$$r := \left\lfloor \frac{\alpha(n-t) + 2tg + \sqrt{\Delta_3}}{2(t^2 - \alpha n)} \right\rfloor + 1$$

and

$$s := \left\lfloor \frac{rt - \frac{\alpha}{2} - g - \sqrt{\Delta_1}}{\alpha} \right\rfloor + 1$$

in Algorithm 3.6 produces a list of s codewords of within distance $n - t$ of any received word $y \in \mathbb{F}^n$, where

$$\Delta_1 = (t^2 - \alpha n)r^2 + (\alpha t - \alpha n - 2tg)r + \frac{\alpha^2}{4} + g^2 - \alpha g > 0$$

as in Lemma 3.7 and

$$\Delta_3 := \alpha^2 \left((n-t)^2 - 4gn\right) + 4\alpha gn \left(t+g\right).$$

Proof. We use the notation

$$s_1 := \frac{rt - \frac{\alpha}{2} - g - \sqrt{\Delta_1}}{\alpha},$$

$$s_2 := \frac{rt - \frac{\alpha}{2} - g + \sqrt{\Delta_1}}{\alpha},$$

and

$$\Delta_2 := \alpha n \left(\alpha n + \alpha^2 + 4g^2 - 4\alpha g - 2t\left(\alpha - 2g\right)\right).$$

as in the statement of Lemma 3.7.

Notice that $s = \lfloor s_1 \rfloor + 1$. We claim that $s_2 - s_1 > 1$ so that $s_1 < s < s_2$. To see this, observe that $s_2 - s_1 = \frac{2\sqrt{\Delta_1}}{\alpha}$. Thus, it suffices to show that $\Delta_1 > \frac{\alpha^2}{4}$. Since $\Delta_3 = \text{disc}\left(\Delta_1 - \frac{\alpha^2}{4}\right)$, we have that

$$\Delta_1 - \frac{\alpha^2}{4} = \left(r - \frac{\alpha\left(n-t\right) + 2tg + \sqrt{\Delta_3}}{2(t^2 - \alpha n)}\right)\left(r - \frac{\alpha\left(n-t\right) + 2tg - \sqrt{\Delta_3}}{2(t^2 - \alpha n)}\right).$$

By the choice of r, it follows that $\Delta_1 - \frac{\alpha^2}{4} > 0$. Therefore, $s_1 < s < s_2$ as claimed.

We next check conditions (a) and (b) of Lemma 3.7(2). For condition (a), we note that

$$s\alpha \leq rt + \frac{\alpha}{2} - g - \sqrt{\Delta_1} < rt - g < rt$$

since $\sqrt{\Delta_1} > \frac{\alpha}{2}$ from above. Condition (b) holds, because

$$\Delta_3 - \Delta_2 = \alpha^2 \left(t^2 - \alpha n\right) > 0.$$

Now applying Lemma 3.7, we see that r and s are valid parameters for the Guruswami-Sudan algorithm. □

Note that the interpolation step may be thought of as polynomial reconstruction, as in [38].

3.4 Examples

In this section, examples are given to illustrate Theorem 3.1.

Example 3.8. *Consider the Hermitian curve of genus* 28 *defined by* $y^8+y=x^9$ *over* $\mathbb{F}_{64}$ *and the code* $C_{\mathcal{L}}(D, 43P_\infty)$ *where* $D := Q_1 + \cdots + Q_{512}$ *is the sum of the* 512 $\mathbb{F}_{64}$*-rational points on the curve other than* P_∞*.*

Let $t = 421$*. Using the parameter choices in Algorithm 3.6, we have* $r = 1$ *and the number of solutions to the reconstruction problem is bounded by* $s = \left\lfloor \frac{(1(421)-1)-28}{43} \right\rfloor = 9$*. Hence, we are guaranteed that there are at most* 9 *codewords within distance* $n - t = 91$ *of a received word* $w \in \mathbb{F}_{64}^{512}$*. By Theorem 3.1, we see that taking* $r = 1$ *and* $s = 1$ *is possible. Thus, applying Algorithm 3.6 with these parameter choices we see that there is a unique codeword within distance* 91 *of* w*. Recall that* C *corrects any* $\left\lfloor \frac{d-1}{2} \right\rfloor$ *erros. In this example, we know that there is a unique codeword within distance* 91 *since* $C_{\mathcal{L}}(D, 43P_\infty)$ *has minimum distance* 469 *(according to [42]).*

Now consider the code $C_{\mathcal{L}}(Q_1 + \cdots + Q_{512}, 217P_\infty)$ *on the same curve. Suppose* $w \in \mathbb{F}_{64}^{512}$ *is a received word, and set* $t = 337$*. By Theorem 3.1, one can take* $r = 24$ *and* $s = 36$ *in the Guruswami-Sudan list decoding algorithm. Applying the algorithm with these parameter choices enables one to work with a degree (at most)* 36 *interpolating polynomial and yields a list of at most* 36 *words which agree with* w *in at least* 337 *places. The original parameter choices give an upper bound of* $s = 83$ *on the number of such words.*

Example 3.9. *Consider the code* $C_{\mathcal{L}}(Q_1 + \cdots + Q_{125}, 58P_\infty)$ *on the Hermitian curve of genus* 10 *defined by* $y^5 + y = x^6$ *over* $\mathbb{F}_{25}$*; here* $Q_1, \ldots, Q_{125}$ *the* 125 *distinct* $\mathbb{F}_{25}$*-rational points on the curve other than* P_∞*. Let* $t = 88$*. The typical parameters in Algorithm 3.6 are* $r = 19$ *and* $s = 28$*. According to Theorem 3.1, we can instead take* $r = 9$ *and* $s = 12$*.*

Hence, there are at most 12 *codewords which agree with a received word* $w \in \mathbb{F}_{25}^{125}$ *in at least* 88 *places (as opposed to at most* 28 *which one might expect given by the original parameter choices in the algorithm).*

3.5 List Decoding AG Codes Over Rings

In this section, we shift our focus to algebraic geometry codes over rings. Codes over rings (that are not necessarily fields) have attracted much attention since the breakthrough created by $\mathbb{Z}_4$-linear codes in [8] and [20]. In the mid-1990's, Judy Walker defined algebraic geometry codes over a local Artinian ring [35] and later, along with Kathy Bartley, presented a list decoding algorithm for these codes in [5]. Previously, Marc Armand ([1] and [2]) studied list decoding of Reed-Solomon codes over rings. In the definition of these codes, the divisors that are used are Cartier divisors, rather than Weil divisors. However the decoding results parallel those of Guruswami and Sudan. As a result we are able to make an analogous improvement to the parameters in the list decoding algorithm that Bartley and Walker provide. We begin with the necessary background and notation to define algebraic geometry codes over rings. Further discussion of the decoding algorithm in the ring setting can be found in [4].

Let A be a local Artinian ring with maximal ideal $\mathfrak{m}$. Let $A/\mathfrak{m}$ be the finite residue field.

Definition 3.10. *A linear code* C *of length* n *over* A *is an* A*-submodule of* A^n*.*

Let $\mathbf{X}$ be a curve over A, by which we mean $\mathbf{X}$ is a scheme, and let $\mathcal{O}_{\mathbf{X}}$ denote its associated valuation ring. The parallel of the Riemann-Roch space in the ring setting is the A-module $\gamma(X, \mathcal{O}_X(G))$ where G is a divisor on $\mathbf{X}$. Let $\alpha \in \mathbb{Z}^+$. Let $P_1, \ldots, P_n, P$ be closed points corresponding to distinct A-points $Z_1, \ldots, Z_n, Z$ on a curve $\mathbf{X}$ of genus g. Let $\mathcal{Z} = \{Z_1, \ldots, Z_n\}$.

The evaluation map is defined by

$$\gamma : \oplus_{i=1}^{n} \Gamma(Z_i, \mathcal{O}_{\mathbf{X}}(\alpha Z)|_{Z_i}) \longrightarrow A^n$$

where $2g - 2 < \alpha < n$.

The one-point algebraic geometry code is $C_{\mathcal{L}}(\mathcal{Z}, \mathcal{O}_{\mathbf{X}}(\alpha Z)) := \mathrm{Im}\ \gamma$.

Bartley and Walker adapted the Guruswami-Sudan algorithm in order to accommodate AG codes over rings. As in the previous list decoding algorithms, their algorithm consists of three steps: initialization, interpolation, and factorization.

Algorithm 3.11 (Bartley-Walker Algorithm)**.**

Input: received word $w \in A^n$, α, agreement parameter t.

Assumptions: $t^2 > \alpha n$.

Let $\Omega := \{f \in \Gamma(\mathbf{X}, \mathcal{O}_{\mathbf{X}}(\alpha Z)) : d(\gamma(f), w) \leq n - t\}$.

0. *Parameter choices: Set*

$$\begin{aligned} r &:= \left\lfloor \tfrac{2gt+\alpha n+\sqrt{(2gt+\alpha n)^2-4(g^2-1)(t^2-\alpha n)}}{2(t^2-\alpha n)} \right\rfloor + 1 \ \textit{and} \\ s &:= \left\lfloor \tfrac{l-g}{\alpha} \right\rfloor. \end{aligned}$$

1. *Interpolation: Find a polynomial $Q[T] \in K[T] \setminus \{0\}$ of degree at most s satisfying the properties*

   (a) $Q(f) \in \Gamma(\mathbf{X}, \mathcal{O}_{\mathbf{X}}((rt-1)Z))$ and

   (b) $Q(f)$ has an r-singularity at Z_i for all i with $f(Z_i) = w_i$
   for all $f \in \Omega$.

2. *Factorization: Find all roots $h \in \Gamma(\mathbf{X}, \mathcal{O}_{\mathbf{X}}(\alpha Z))$ of the polynomial Q. For each such h, if $h(Q_i) = w_i$ for at least t values of i, then add h to the output list.*

Output: $h_1, \ldots, h_s$ such that $d(w, \gamma(h_i)) \leq n - t$.

As seen for the Guruswami-Sudan algorithm, it is advantageous to choose the parameters that result in a smaller degree interpolating polynomial Q and yield a better bound s on the list size of the output. We can make improvements by looking at the constraints and viewing them as quadratic polynomials. With these additional observations, it is possible to choose an r and s that will satisfy the two conditions above, while giving a smaller upper bound on the list size.

Next, we indicate how Lemma 3.7 can be used in conjunction with Algorithm 3.11 to obtain a better bound on the list size.

Theorem 3.2. *Consider the AG code $C := C_{\mathcal{L}}(\mathcal{Z}, \mathcal{O}_{\mathbf{X}}(\alpha Z)) \subseteq A^n$ defined using a curve X of genus g over a local Artinian ring A and $D = Z_1 + \cdots + Z_n$.*
Suppose (i) $t^2 > \alpha n$ and (ii) either $\alpha < 2g$ or $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)$. Then taking

$$r := \left\lfloor \frac{\alpha(n-t) + 2tg + \sqrt{\Delta_3}}{2(t^2 - \alpha n)} \right\rfloor + 1 \text{ and } s := \left\lfloor \frac{rt - \frac{\alpha}{2} - g - \sqrt{\Delta_1}}{\alpha} \right\rfloor + 1$$

in Algorithm 3.11 produces a list of s codewords of within distance $n - t$ of any received word $w \in A^n$, where

$$\Delta_1 = (t^2 - \alpha n) r^2 + (\alpha t - \alpha n - 2tg) r + \frac{\alpha^2}{4} + g^2 - \alpha g$$

as in Lemma 3.7 and

$$\Delta_3 := \alpha^2 \left((n-t)^2 - 4gn\right) + 4\alpha g n \,(t + g)\,.$$

Proof. We use the notation

$$s_1 := \frac{rt - \frac{\alpha}{2} - g - \sqrt{\Delta_1}}{\alpha},$$

$$s_2 := \frac{rt - \frac{\alpha}{2} - g + \sqrt{\Delta_1}}{\alpha},$$

and

$$\Delta_2 := \alpha n \left(\alpha n + \alpha^2 + 4g^2 - 4\alpha g - 2t\,(\alpha - 2g)\right).$$

as in the statement of Lemma 3.7.

Notice that $s = \lfloor s_1 \rfloor + 1$. We claim that $s_2 - s_1 > 1$ so that $s_1 < s < s_2$. To see this, observe that $s_2 - s_1 = \frac{2\sqrt{\Delta_1}}{\alpha}$. Thus, it suffices to show that $\Delta_1 > \frac{\alpha^2}{4}$. Since

$\Delta_3 = \operatorname{disc}\left(\Delta_1 - \frac{\alpha^2}{4}\right)$, we have that

$$\Delta_1 - \frac{\alpha^2}{4} = \left(r - \frac{\alpha\,(n-t) + 2tg + \sqrt{\Delta_3}}{2(t^2 - \alpha n)}\right)\left(r - \frac{\alpha\,(n-t) + 2tg - \sqrt{\Delta_3}}{2(t^2 - \alpha n)}\right).$$

By the choice of r, it follows that $\Delta_1 - \frac{\alpha^2}{4} > 0$. Therefore, $s_1 < s < s_2$ as claimed.

We next check conditions (a) and (b) of Lemma 3.7(2). For condition (a), we note that

$$s\alpha \leq rt + \frac{\alpha}{2} - g - \sqrt{\Delta_1} < rt - g < rt$$

since $\sqrt{\Delta_1} > \frac{\alpha}{2}$ from above. Condition (b) holds, because

$$\Delta_3 - \Delta_2 = \alpha^2\left(t^2 - \alpha n\right) > 0.$$

Now applying Lemma 3.7, we see that r and s are valid parameters for the Bartley-Walker algorithm. □

3.6 List Decoding AG Codes Using Places of Higher Degree

Often we consider AG codes $C_{\mathcal{L}}(D, \alpha P)$ where P is a place of degree one. Both [40] and [26] have shown that by using places of higher degree, codes with better parameters may be obtained. In this section, we provide a list decoding algorithm for AG codes $C_{\mathcal{L}}(D, \alpha P)$ where P is a place of degree $r > 1$.

Let X be a curve of genus g over a finite field $\mathbb{F}$ and let $D = P_1 + \cdots + P_n$ be a divisor. Consider the code $C_{\mathcal{L}}(D, \alpha P)$ where P is a place of degree $r > 1$. Suppose that $w \in \mathbb{F}^n$ is a received word and t is an agreement parameter. Our goal is to find

$$\Omega := \left\{f \in \mathcal{L}(\alpha P) : d\left(\left(f(P_1), \ldots, f(P_n)\right), w\right) \leq n - t\right\}.$$

We do this by

1. Determining an interpolating polynomial $Q(T) \in K[T] \setminus \{0\}$ such that

$$f \in \Omega \Rightarrow Q(f) = 0,$$

and

2. Finding roots of $Q(T)$.

   We first verify that the interpolating polynomial evaluated at $f \in \Omega$ is zero.

   We know

$$Q(T) = u_0 + u_1 T + u_2 T^2 + \cdots + u_s T^s$$

with $u_i \in K$ for some $s \in \mathbb{Z}^+$.

Suppose $f \in \mathcal{L}(\alpha P)$.Then

$$Q(f) = u_0 + u_1 f + u_2 f^2 + \cdots + u_s f^s \in K.$$

To guarantee that $Q(f) = 0$ for all $f \in \Omega$, require that the number of zeros of $Q(f)$ be greater than the number of poles.

To restrict the number of poles, we require that $Q(f) \in \mathcal{L}(lP)$. Note that,

$$\begin{aligned} v_P\left(Q(f)\right) &= v_P\left(u_0 + u_1 f + u_2 f^2 + \cdots + u_s f^s\right) \\ &\geq \min_i v_P\left(u_i f^i\right) \\ &= \min_i v_P\left(u_i\right) + i v_P\left(f\right) \\ &\geq \min_i v_P\left(u_i\right) - i\alpha. \end{aligned}$$

Hence, we require that

$$v_P\left(u_i\right) - i\alpha \geq -l \qquad \forall i, 0 \leq i \leq s,$$

i.e.,

$$u_i \in \mathcal{L}\left(\left(l - i\alpha\right) P\right) \qquad \forall i, 0 \leq i \leq s.$$

Recall that $u_i \in \mathcal{L}\left(\left(l - i\alpha\right) P\right)$. Since

$$\mathcal{L}\left(\left(l - s\alpha\right) P\right) \subseteq \cdots \subseteq \mathcal{L}\left(\left(l - 2\alpha\right) P\right) \subseteq \mathcal{L}\left(\left(l - \alpha\right) P\right) \subseteq \mathcal{L}\left(lP\right),$$

one can build a basis for $\mathcal{L}(lP)$ as follows:

$$\begin{aligned}
&\phi_1, \ldots, \phi_{\dim \mathcal{L}((l-s\alpha)P)} \text{ is a basis for } \mathcal{L}((l-s\alpha)P)\\
&\phi_{\dim \mathcal{L}((l-s\alpha)P)+1} \cdots, \phi_{\dim \mathcal{L}((l-(s-1)\alpha)P)} \text{ is a basis for } \mathcal{L}((l-(s-1)\alpha)P)\\
&\qquad\quad\vdots\\
&\phi_{\dim \mathcal{L}((l-2\alpha)P)+1} \cdots, \phi_{\dim \mathcal{L}((l-\alpha)P)} \text{ is a basis for } \mathcal{L}((l-\alpha)P)\\
&\phi_{\dim \mathcal{L}((l-\alpha)P)+1} \cdots, \phi_{\dim \mathcal{L}(lP)} \text{ is a basis for } \mathcal{L}(lP).
\end{aligned}$$

Then

$$u_i = \sum_{j=1}^{\dim \mathcal{L}((l-i\alpha)P)} a_{ij}\phi_j$$

for some $a_{ij} \in \mathbb{F}$.

Next, we require certain zeros of $Q(f)$ for $f \in \Omega$.

Assume $f \in \Omega$. We want to have $v_{P_k}(Q(f)) \geq M$ for all k such that $f(P_k) = w_k$. To accomplish this we "shift". The basic idea is $b \in \mathbb{F}$ is a zero of $h(T) \in \mathbb{F}[T]$ of multiplicity at least M if and only if

$$h(T) = (T-b)^M g(T) \text{ if and only if } h(T+b) = T^M g(T+b).$$

Rather than requiring that $v_{P_k}(Q(f)) \geq M$ for all k such that $f(P_k) = w_k$, we require

$$v_{P_k}(Q(f+w_k)) \geq M$$

for all k such that $f(P_k) = 0$.

Lemma 3.12. *Given any point $P_k \in suppD$, there exist $\psi_{1k}, \ldots, \psi_{\dim\mathcal{L}((l-i\alpha)P)k}$ with*

$$\langle \psi_{1k}, \ldots, \psi_{\dim\mathcal{L}((l-i\alpha)P)k} \rangle = \langle \phi_1, \ldots, \phi_{\dim\mathcal{L}((l-i\alpha)P)} \rangle$$

and

$$v_{P_k}(\psi_{jk}) \geq j-1.$$

By the lemma,

$$\begin{array}{rcl} Q(T) & = & \sum_{i=0}^{s} \sum_{j=1}^{\dim \mathcal{L}((l-i\alpha)P)} a_{ij}\phi_j T^i \\ & = & \sum_{i=0}^{s} \sum_{j=1}^{\dim \mathcal{L}((l-i\alpha)P)} a_{ij} \sum_{m=1}^{\dim \mathcal{L}((l-i\alpha)P)} b_{jkm}\psi_{mk}T^i \end{array}$$

and so

$$Q(T+w_k) = \sum_{i=0}^{s} \sum_{j=1}^{\dim \mathcal{L}((l-i\alpha)P)} a_{ij}^{(k)}\psi_{jk}T^i.$$

Hence,

$$\begin{array}{rcl} v_{P_k}\left(Q(f+w_k)\right) & \geq & \min\left\{v_{P_k}\left(\psi_{jk}f^i\right) : i,j \text{ with } a_{ij}^{(k)} \neq 0\right\} \\ & = & \min\left\{v_{P_k}\left(\psi_{jk}\right) + iv_{P_k}\left(f\right) : i,j \text{ with } a_{ij}^{(k)} \neq 0\right\} \\ & \geq & \min\left\{j-1+i : i,j \text{ with } a_{ij}^{(k)} \neq 0\right\}. \end{array}$$

So we require that $a_{ij}^{(k)} = 0$ for all $i+j-1 < M$.

Next we verify that such an interpolating polynomial exists. To ensure that $u_s \neq 0$, assume $\dim \mathcal{L}\left(\left(l-s\alpha\right)P\right) \geq \left(l-s\alpha\right)r + 1 - g \geq 1$. Take

$$s := \left\lfloor \frac{lr-g}{\alpha r} \right\rfloor.$$

Then the number of coefficients of Q is

$$\sum_{i=0}^{s} \dim \mathcal{L}\left(\left(l-i\alpha\right)P\right) \geq \frac{\left(lr+1-g\right)^2 - 1}{2\alpha r}.$$

Choose M so that

$$\frac{\left(lr+1-g\right)^2 - 1}{2\alpha r} \geq n\binom{M+1}{2}.$$

If $rt^2 - \alpha n > 0$, take

$$M = 1 + \left\lfloor \frac{2r^2t - 2rt + 2grt + \alpha rn + \sqrt{\Delta}}{2(r^2t^2 - \alpha rn)} \right\rfloor$$

where

$$\Delta = (2r^2t - 2rt + 2grt + \alpha rn)^2 - 4(r^2t^2 - \alpha rn)(g^2 - 2g + 2rg - 2r + r^2).$$

We now have all of the necessary ingredients for an algorithm for list decoding one-point AG codes defined by places of higher degree.

Algorithm 3.13. *Let $C := C_{\mathcal{L}}(P_1 + \cdots + P_n, \alpha P)$ be an $[n, k, d]$ code defined using a curve X of genus g over $\mathbb{F}_q$ and K be the function field of X and P a place of degree r.*

The input consists of a received word $w \in \mathbb{F}^n$, the code length n, α, and an agreement parameter t such that $rt^2 - \alpha n > 0$.

0. Initialization: Fix parameters

$$\begin{array}{rcl} M & := & 1 + \left\lfloor \frac{2r^2t - 2rt + 2grt + \alpha rn + \sqrt{\Delta}}{2(r^2t^2 - \alpha rn)} \right\rfloor, \\ s & := & \left\lfloor \frac{lr - g}{\alpha r} \right\rfloor, \textit{and} \\ l & := & Mt - 1 \end{array}$$

where $\Delta = (2r^2t - 2rt + 2grt + \alpha rn)^2 - 4(r^2t^2 - \alpha rn)(g^2 - 2g + 2rg - 2r + r^2)$.

1. Interpolation: Find $Q(T) \in K[T] \setminus \{0\}$ as described above.

2. Factorization: Find all roots $f \in \Omega$ of $Q(T)$ in order to output a list of functions $h \in \mathcal{L}(\alpha P)$ such that $d(ev(h), w) \leq n - t$.

Finally, we verify that h is a root of $Q(T)$ for each $h \in \mathcal{L}(\alpha P)$ with $d(ev(h), w) \leq n - t$. Suppose $h \in \mathcal{L}(\alpha P)$ is such that $h(P_i) = w_i$ for at least t values of $i \in \{1, \ldots, n\}$. By construction, $Q(h) \in \mathcal{L}(lP)$. However,

$$\#\text{ zeros of } Q(h) \geq \sum_{i=1}^{n} v_{P_i}(Q(h)) \geq Mt > l \geq \#\text{ poles of } Q(h).$$

Hence, h is among the roots of $Q(T)$.

CHAPTER 4

DECODING GENERAL AG CODES USING LISTS

4.1 Introduction

In this chapter, we give a minimum distance decoding algorithm for a general algebraic geometry code C by viewing it as a subcode of a one-point code C'. We show that unique decoding in C up to its minimum distance may be achieved by list decoding in C'. As a result, we obtain a decoding algorithm for decoding general algebraic geometry codes up to the minimum distance.

Recall that algebraic geometry codes are defined by using a divisor G on a curve over a finite field $\mathbb{F}$. Often, G is supported by a single $\mathbb{F}$-rational point and the resulting code is a one-point code. Recently, there has been interest in allowing the divisor G to be more general as this can result in superior codes [6, 9, 10, 26, 40, 41]. In particular, one may obtain a code with better parameters by allowing G to be supported by m distinct $\mathbb{F}$-rational points, where $m > 1$ [24, 23, 25, 17, 18, 19]. While multipoint codes may have better parameters than comparable one-point codes on the same curve, most decoding algorithms have been designed specifically for one-point codes. Exceptions are Beelen's adaptation of majority voting [6] and the modification of the Berlekamp-Massey-Sakata Algorithm [31] due to Sakata [30], which decode up to the generalized order bound, a lower bound on the minimum distance. However, these modifications can be quite tedious and the generalized order bound does not agree with the actual minimum distance in general (though it does in certain cases, for example, for two-point Hermitian codes). In this chapter, we see that list decoding in a supercode provides a simple algorithm for decoding a multipoint code up to its minimum distance.

By embedding a multipoint code in a one-point code, we can capitalize on the various improvements to the list decoding algorithms presented such as [28] and [29] without completely retooling the algorithm.

This chapter is organized as follows. Section 4.2 presents a decoding algorithm for general AG codes based on a list decoding algorithm for one-point codes. In Section 4.3, this algorithm is modified to handle multiple one-point supercodes. Finally, examples are provided in Section 4.4.

4.2 A Minimum Distance Decoder for Multipoint Codes Via Lists

In this section, we outline the decoding algorithm for multipoint codes. Note that the point of view taken here can be utilized with any list decoding algorithm for one-point codes. For clarity of exposition, we focus on the Guruswami-Sudan list decoding algorithm as found in [14, Section IV. B.]. In all that follows, K denotes the function field associated with X.

The next result shows that every multipoint code C supported by points $P_1, \ldots, P_m$ is (isometric to one) of the form $C_{\mathcal{L}}(D, a_1 P_1 - \sum_{i=2}^{m} a_i P_i)$.

Lemma 4.1. *Let X be a nonsingular projective curve over a finite field $\mathbb{F}$. Given a multipoint code $C_{\mathcal{L}}(D, G)$ on X, $C_{\mathcal{L}}(D, G)$ is isometric to a subcode of a one-point code $C_{\mathcal{L}}(D, \alpha P)$ on X for some $\mathbb{F}$-rational point P in the support of G.*

Proof. Consider the multipoint code $C := C_{\mathcal{L}}(D, \sum_{i=1}^{m} a_i P_i)$ over $\mathbb{F}_q$. Without loss of generality, we may assume $a_i \in \mathbb{Z}^+$ for all $1 \leq i \leq m$. Since $\mathbb{F}_q$ is finite, the class number of X is finite, and consequently, there exists a rational function f with divisor

$$(f) = \sum_{i=2}^{m} b_i P_i - b_1 P_1$$

where $b_i \geq a_i$ for all i, $2 \leq i \leq m$, and $b_1 := \sum_{i=2}^{m} b_i$ [33, Proposition V.1.3].

The function f defines a vector space isomorphism

$$\begin{array}{rccc} \phi : & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q^n \\ & v & \longmapsto & ev(f) * v \end{array}$$

where

$$ev(f) * v := (f(Q_1) \cdot v_1, \ldots, f(Q_n) \cdot v_n).$$

Since f has no zeros among $Q_1, \ldots, Q_n$, the map ϕ is weight-preserving, and hence, distance-preserving. Restricting ϕ to $C_{\mathcal{L}}\left(D, \sum_{i=1}^m a_i P_i\right)$ induces an isometry ϕ of codes

$$C_{\mathcal{L}}\left(D, \sum_{i=1}^m a_i P_i\right) \overset{\phi}{\cong} C_{\mathcal{L}}\left(D, (a_1 + b_1) P_1 - \sum_{i=2}^m (b_i - a_i) P_i\right)$$

as

$$(fh) \geq \sum_{i=2}^m (b_i - a_i) P_i - (a_1 + b_1) P_1$$

for all $h \in \mathcal{L}\left(\sum_{i=1}^m a_i P_i\right)$. If $a_i = b_i$ for all i, $2 \leq i \leq m$, then the m-point code is actually (isometric to) a one-point code. For this reason, we assume that $b_i > a_i$ for some i, $2 \leq i \leq m$, and hence $\sum_{i=2}^m b_i > \sum_{i=2}^m a_i$. □

Algorithm 4.2. *Let $C := C_{\mathcal{L}}(D, a_1 P_1 - \sum_{i=2}^m a_i P_i)$ be an m-point code over the finite field $\mathbb{F}_q$ where $D := Q_1 + \cdots + Q_n$. Suppose that $w \in \mathbb{F}_q^n$ is a received word in which $\left\lfloor \frac{d(C)-1}{2} \right\rfloor$ or fewer errors have occurred.*

Input: $a_1, \ldots, a_m$, received word $w \in \mathbb{F}_q^n$, agreement parameter $t := n - \left\lfloor \frac{d(C)-1}{2} \right\rfloor$.

Assumptions: $t^2 > a_1 n$.

Set $\Omega := \{f \in \mathcal{L}(a_1 P) : d(ev(f), w) \leq n - t\}$.

0. Fix parameters: Set

$$r := \left\lfloor \frac{2gt + a_1 n + \sqrt{(2gt + a_1 n)^2 - 4\left(g^2 - 1\right)\left(t^2 - a_1 n\right)}}{2\left(t^2 - a_1 n\right)} \right\rfloor + 1$$

and

$$s := \left\lfloor \frac{rt-1-g}{a_1} \right\rfloor .$$

1. *Interpolation: Find a nonzero interpolating polynomial $Q(T) \in K[T]$ where K denotes the function field associated with X, satisfying*

   (a) $Q(f) \in \mathcal{L}((rt-1)P_1)$ for all $f \in \mathcal{L}(a_1P_1)$, and

   (b) $v_{Q_i}(Q(h)) \geq r$ for each $i \in \{1, \ldots, n\}$ with $H(Q_i) = w_i$.

2. *Factorization: Find all roots $h \in \mathcal{L}(a_1P)$ of the polynomial Q. For each such h, if $h(Q_i) = w_i$ for at least t values of i, then add h to Ω. In this way, we find all functions $h \in \mathcal{L}(a_1P_1)$ that possibly give rise to the codewords in $C' := C_{\mathcal{L}}(D, a_1P_1)$ at distance $\left\lfloor \frac{d(C)-1}{2} \right\rfloor$ from w.*

3. *Check for zeros: Compute the order of h at P_i for each h found in Step (3) until the one is found with $v_{P_i}(h) \geq -a_i$ for all i, $2 \leq i \leq m$.*

4. *Decode w as $((h(Q_1), \ldots, h(Q_n))$.*

Output: $((h(Q_1), \ldots, h(Q_n))$, the unique word in C with $d(ev(h), w) \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$.

Remark 4.3. *Steps (1)-(3) of Algorithm 4.2 may be replaced with those of any list decoding algorithm for $C_{\mathcal{L}}(D, a_1P_1)$ which yields Ω as its output.*

The goal of Step (4) may be achieved via parity checks [37]. Specifically, one could determine the additional parity checks $v_1, \ldots, v_r$ that words in C' must satisfy to be in the subcode C. Then, for each h found in Step (3), compute $ev(h) \cdot v_i$, $1 \leq i \leq r$, until an h is found satisfying all r checks.

Remark 4.4. *It should be noted that if we allow the divisor G to contain points of higher order, we can still decode using a subcode. In this setting, the analysis parallels that above.*

For the purpose of decoding it is sufficient to consider the code $\phi(C)$. Suppose that $w \in \mathbb{F}_q^n$ is received using the code C and that E errors have occurred, where $E \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$. We may identify w and $ev(f) * w$ via the map ϕ. Then $ev(f) * w$ is treated as a received word using $\phi(C)$. Since ϕ is distance preserving, $E \leq \left\lfloor \frac{d(\phi(C))-1}{2} \right\rfloor$ and so there is a unique nearest codeword $ev(h) \in \phi(C)$ to $ev(f) * w$. Then $ev(f^{-1}h)$ is the unique codeword in C nearest w.

In the literature, m-point codes are often given in the form $C_{\mathcal{L}}(D, \sum_{i=1}^{m} a_i P_i)$ with $a_i \in \mathbb{Z}^+$ for all $1 \leq i \leq m$. The following examples illustrate an isometry ϕ described above for commonly studied multipoint codes.

Example 4.5. *In this example, we consider two-point Hermitian codes. Recall that the Hermitian curve may be defined by $y^q + y = x^{q+1}$ over $\mathbb{F}_{q^2}$. Since the automorphism group of this curve is doubly-transitive, we may restrict our attention to a code of the form $C := C_{\mathcal{L}}\left(D, aP_\infty + bP_{00}\right)$. Suppose $a, b \in \mathbb{Z}^+$. Then multiplication by*

$$f := y^{\left\lceil \frac{b}{q+1} \right\rceil}$$

induces a vector space isomorphism

$$\mathcal{L}\left(D, aP_\infty + bP_{00}\right) \cong \mathcal{L}\left(\left(a + \left\lceil \frac{b}{q+1} \right\rceil (q+1)\right) P_\infty + \left(b - \left\lceil \frac{b}{q+1} \right\rceil (q+1)\right) P_{00}\right),$$

because

$$(y) = (q+1)\left(P_{00} - P_\infty\right).$$

Hence,

$$C \cong C_{\mathcal{L}}\left(D, \left(a + \left\lceil \frac{b}{q+1} \right\rceil (q+1)\right) P_\infty - \left(\left\lceil \frac{b}{q+1} \right\rceil (q+1) - b\right) P_{00}\right).$$

Example 4.6. Now consider an m-point Hermitian code

$$C := C_{\mathcal{L}}\left(D, a_1 P_\infty + \sum_{i=2}^{m} a_i P_{\alpha\beta_i}\right)$$

supported by collinear points $P_\infty, P_{\alpha\beta_2}, \ldots, P_{\alpha\beta_m}$ where $a_i \in \mathbb{Z}^+$ for all i, $1 \leq i \leq m$ and $2 \leq m \leq q+1$. Such codes were studied in [22] where the authors show that if

$$\tau_{\alpha\beta_i} := y - \beta_i - \alpha^q(x - \alpha)$$

then

$$\left(\tau_{\alpha\beta_i}\right) = (q+1)\left(P_{\alpha\beta_i} - P_\infty\right).$$

Thus we can take

$$f = \prod_{i=2}^{m} \tau_{\alpha\beta_i}^{\lceil \frac{a_i}{q+1} \rceil}.$$

The multiplication by f induces a vector space isomorphism

$$\mathcal{L}\left(a_1 P_\infty + \textstyle\sum_{i=2}^{m} a_i P_{\alpha\beta_i}\right) \cong$$

$$\mathcal{L}\left(\left(a_1 + (q+1)\textstyle\sum_{i=2}^{m} \left\lceil \frac{a_i}{q+1} \right\rceil\right) P_\infty + \textstyle\sum_{i=2}^{m} \left(a_i - \left\lceil \frac{a_i}{q+1} \right\rceil (q+1)\right) P_{\alpha\beta_i}\right)$$

and an isometry of codes

$$\begin{aligned} C &\cong C_{\mathcal{L}}\left(D, \left(a_1 + (q+1)\textstyle\sum_{i=2}^{m} \left\lceil \frac{a_i}{q+1} \right\rceil\right) P_\infty - \textstyle\sum_{i=2}^{m} \left(\left\lceil \frac{a_i}{q+1} \right\rceil (q+1) - a_i\right) P_{\alpha\beta_i}\right) \\ &\subseteq C_{\mathcal{L}}\left(D, \left(a_1 + (q+1)\textstyle\sum_{i=2}^{m} \left\lceil \frac{a_i}{q+1} \right\rceil\right) P_\infty\right) \end{aligned}$$

since

$$\begin{aligned} &\mathcal{L}\left(\left(a_1 + (q+1)\textstyle\sum_{i=2}^{m} \left\lceil \frac{a_i}{q+1} \right\rceil\right) P_\infty + \textstyle\sum_{i=2}^{m} \left(a_i - \left\lceil \frac{a_i}{q+1} \right\rceil (q+1)\right) P_{\alpha\beta_i}\right) \\ &\subseteq \mathcal{L}\left(\left(a_1 + (q+1)\textstyle\sum_{i=2}^{m} \left\lceil \frac{a_i}{q+1} \right\rceil\right) P_\infty\right). \end{aligned}$$

Example 4.7. In this example, let $C := C_{\mathcal{L}}(D, aP_\infty + bP_{00})$ be a two-point Suzuki code where $a, b \in \mathbb{Z}^+$. The Suzuki curve is defined over $\mathbb{F}_q$ by the equation

$$y^q - y = x^{q_0}(x^q - x)$$

where $q_0 = 2^n$, $q = 2^{2n+1}$, and $n \in \mathbb{Z}^+$. Let

$$w := y^{\frac{q}{q_0}} x^{\frac{q}{q_0^2}+1} + \left(y^{\frac{q}{q_0}} - x^{\frac{q}{q_0^2}+1}\right)^{\frac{q}{q_0}}.$$

Since

$$(w) = \left(q + \frac{q}{q_0} + 1\right)(P_{00} - P_\infty)$$

as shown in [16], multiplication by

$$f := w^{\left\lceil \frac{b}{q + \frac{q}{q_0} + 1} \right\rceil}$$

gives rise to an isomorphism of Riemann-Roch spaces and consequently an isometry of codes

$$C \cong C_{\mathcal{L}}\left(D, \alpha P_{\infty} - \beta P_{00}\right)$$

where

$$\alpha = a + \left\lceil \frac{b}{q + \frac{q}{q_0} + 1} \right\rceil \left(q + \frac{q}{q_0} + 1\right) \text{ and } \beta = \left\lceil \frac{b}{q + \frac{q}{q_0} + 1} \right\rceil \left(q + \frac{q}{q_0} + 1\right) - b.$$

4.3 A Minimum Distance Decoder for Multipoint Codes Using Lists, Multiple Embeddings, and GCD

In this section, we discuss a modification of Algorithm 4.2 in which a multipoint code is embedded in multiple one-point codes and the interpolating polynomial is obtained as a greatest common divisor. This idea was inspired by [3].

Consider a multipoint code $C := C_{\mathcal{L}}\left(D, \sum_{i=1}^{m} a_i P_i\right)$ where $a_i \in \mathbb{Z}$. Given any function f whose divisor is supported only by points among $P_1, \ldots, P_m$, multiplication by f induces a vector space isomorphism

$$\mathcal{L}\left(\sum_{i=1}^{m} a_i P_i\right) \cong \mathcal{L}\left(\sum_{i=1}^{m} \left(a_i - v_{P_i}(f)\right) P_i\right)$$

and an isometry of codes

$$C \overset{\phi}{\cong} C_{\mathcal{L}}\left(D, \sum_{i=1}^{m} \left(a_i - v_{P_i}(f)\right) P_i\right).$$

Hence, for each such function f with $v_{P_j}(f) < a_j$ for exactly one j, $1 \leq j \leq m$, C is isometric to a subcode of the one-point code $C_{\mathcal{L}}\left(D, \left(a_j - v_{P_j}(f)\right) P_j\right)$; that is,

$$\phi(C) \subseteq C_{\mathcal{L}}\left(D, \left(a_j - v_{P_j}(f)\right) P_j\right).$$

To emphasize that the embedding is induced by

$$f \in \mathcal{L}\left(a_j P_j - \sum_{\substack{i=1 \\ i \neq j}}^{m} a_i P_i\right)$$

we sometimes write ϕ_j instead of ϕ. The following algorithm exploits these multiple embeddings.

Algorithm 4.8. *Let $C := C_{\mathcal{L}}\left(D, \sum_{i=1}^{m} a_i P_i\right)$ be an m-point code over the finite field $\mathbb{F}_q$ where $D := Q_1 + \cdots + Q_n$. Suppose that $w \in \mathbb{F}_q^n$ is a received word in which $\left\lfloor \frac{d(C)-1}{2} \right\rfloor$ or fewer errors have occurred.*

Input: $a_1, \ldots, a_m$, received word $w \in \mathbb{F}_q^n$, agreement parameter $t := n - \frac{d(C)-1}{2}$.

0. *Embedding: Choose a nonempty subset $J \subseteq \{1, \ldots, m\}$. For each $j \in J$, find a one-point code $C_j := C_{\mathcal{L}}\left(D, \left(a_j - b_{jj}\right) P_j\right)$ such that*

$$C \overset{\phi_j}{\cong} C_{\mathcal{L}}\left(D, \left(a_j - b_{jj}\right) P_j - \sum_{\substack{1 \leq i \leq m \\ i \neq j}} (b_{ij} - a_i) P_i\right) \subseteq C_j$$

is the embedding induced by a rational function f_j with $v_{P_i}\left(f_j\right) = b_{ij}$ for all $1 \leq i \leq m$, $b_{ij} \geq a_i$ for all $i \neq j$, $b_{jj} < a_j$, and $t^2 > \left(a_j - b_{jj}\right) n$.

1. *Fix parameters: For each C_j, fix parameters as in Step (1) of Algorithm 4.2.*

2. *Interpolation: For each C_j, find a nonzero interpolating polynomials $H_j(T) \in K[T]$ as in Step (2) of Algorithm 4.2. Set*

$$Q(T) := gcd\left\{Q_j(T) : j \in J\right\}$$

where $Q_j(T) = H_j\left(f_j T\right)$.

3. *Factorization: Find the roots of $Q(T)$ as in the standard factorization step. In this way, we find all functions $h \in \mathcal{L}\left(\sum_{i=1}^{m} a_i P_i\right)$ that possibly give rise to the codewords in C at distance $\left\lfloor \frac{d-1}{2} \right\rfloor$ from w.*

4. *Check for zeros: Compute the order of h at P_i for each h found in Step (5) until the one is found with $v_{P_i}(h) \geq -a_i$ for all $i \notin J$.*

5. *Decode w as $\left(h\left(Q_1\right), \ldots, h\left(Q_n\right)\right)$.*

Output: $\left(\left(h(Q_1), \ldots, h(Q_n)\right)\right)$, the unique word in C with $d\left(ev(h), w\right) \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$.

Theorem 4.1. *Given a multipoint code $C := C_{\mathcal{L}}\left(D, \sum_{i=1}^{m} a_i P_i\right)$ as above, Algorithm 4.8 provides a minimum distance decoder for C.*

Proof. Suppose that w is a received word in which at most $\left\lfloor \frac{d(C)-1}{2} \right\rfloor$ errors have occured. Then there exists a unique codeword $ev(h)$ that is the transmitted word resulting in the received word w. We must show that the output of Algorithm 4.8 is $ev(h)$.

Assume $h \in \mathcal{L}\left(\sum_{i=1}^{m} a_i P_i\right)$ and $d(ev(h), w) \le \left\lfloor \frac{d(C)-1}{2} \right\rfloor$. We claim that h is a root of $H_j(f_j T)$ for all $j \in J$. Note that among the roots of $H_j(T)$ are elements of

$$\Omega'_j := \left\{ f \in \mathcal{L}((a_j - b_{jj})P_j) : d(ev(f), \phi_j(w)) \le \left\lfloor \frac{d\left(C\right) - 1}{2} \right\rfloor \right\}.$$

We prove that $f_j h \in \Omega'_j$ for all $j \in J$. Let $j \in J$. Then one can check that $f_j \in \mathcal{L}((a_j - b_{jj})P_j)$ by considering divisors of h and f_j. Since $ev(f_j h) = \phi_j(ev(h))$ and ϕ_j is distance preserving,

$$d(ev(f_j h), \phi_j(w)) = d(ev(h), w) \le \left\lfloor \frac{d\left(C\right) - 1}{2} \right\rfloor.$$

Hence, h is a root of $Q(T)$ and so will be found using Algorithm 4.8. □

4.4 Examples

Example 4.9. *Let X denote the Hermitian curve $y^8 + y = x^9$ over $\mathbb{F}_{64}$. Consider the two-point code $C := C_{\mathcal{L}}(D, 344P_{00} - 8P_{\infty})$ where $D := P_1 + \cdots + P_{511}$ is the sum of all $\mathbb{F}_{64}$-rational points other than P_{00} and P_{∞}. Then C is a $[511, 309, 175]$ code and so it can correct 87 errors.*

We can embed this code into $C' := C_{\mathcal{L}}(D, 344P_{00})$, a $[511, 309, 168]$ code which can correct 83 errors. Suppose that $w \in \mathbb{F}_{64}^{511}$ is a received word in which 87 errors have occurred. Applying the Guruswami-Sudan Algorithm with improved parameter choices to C' produces a list of at most 21 functions $h_1, \ldots, h_{21} \in \mathcal{L}(344P_{00})$.

Among this list, there is a unique h_i with $v_{P_\infty}(h_i) \ge 8$. Therefore, we can decode w as $(h_i(P_1), \ldots, h_i(P_n))$.

Example 4.10. *Let X denote the Hermitian curve $y^q + y = x^{q+1}$ over $\mathbb{F}_{q^2}$. Let $D := \sum_{i=1}^{q^3-q} P_i$ be the sum of points of degree one other than the point P_∞ and the q points of the form P_{0b}. Consider the multipoint Hermitian code*

$$C := C_{\mathcal{L}}\left(D, \alpha P_\infty + \beta P_{00} - \sum_{\substack{b \in \mathbb{F}_{q^2} \setminus \{0\} \\ b^q + b = 0}} P_{0b} \right)$$

where $\alpha, \beta \in \mathbb{Z}^+$.

We can embed this code into $C_1' := C_{\mathcal{L}}(D, (\alpha + r_1(q+1))P_\infty)$ where $r_1 \in \mathbb{Z}^+$ such that $r_1(q+1) > \beta$. We identify an isomorphic multipoint code

$$C_1 := C_{\mathcal{L}}\left(D, (\alpha + r_1(q+1))P_\infty + (\beta - r_1(q+1))P_{00} - \sum_{\substack{b \in \mathbb{F}_{q^2} \setminus \{0\} \\ b^q + b = 0}} P_{0b} \right)$$

by multiplying by the function y^{r_1}.

Alternatively, we can embed this code into $C_2' := C_{\mathcal{L}}(D, (\beta + r_2(q+1))P_{00})$ where $r_2 \in \mathbb{Z}^+$ such that $r_2(q+1) > \alpha$. We identify an isomorphic multipoint code

$$C_2 := C_{\mathcal{L}}\left(D, (\alpha - r_2(q+1))P_\infty + (\beta + r_2(q+1))P_{00} - \sum_{\substack{b \in \mathbb{F}_{q^2} \setminus \{0\} \\ b^q + b = 0}} P_{0b} \right)$$

by multiplying by the function $\frac{1}{y^{r_2}}$.

Additionally, we can embed this code into $C_3' := C_{\mathcal{L}}(D, (\alpha + (\beta + 1)q)P_\infty)$. We identify an isomorphic multipoint code

$$C_3 := C_{\mathcal{L}}\left(D, (\alpha + (\beta + 1)q)P_\infty - P_{00} - \sum_{\substack{b \in \mathbb{F}_{q^2} \setminus \{0\} \\ b^q + b = 0}} (2 + \beta)P_{0b} \right)$$

by multiplying by the function $x^{(\beta+1)}$.

For C'_1, C'_2, and C'_3 we find nozero interpolating polynomials $H_1(T)$, $H_2(T)$, and $H_3(T)$. Then we find $Q(T) := \gcd(Q_1(T), Q_2(T), Q_3(T))$ where

$$Q_1(T) := H_1\left(y^{r_1}T\right),$$

$$Q_2(T) := H_2\left(\frac{1}{y^{r_2}}T\right),$$

and

$$Q_3(T) := H_3\left(x^{\beta+1}T\right).$$

We now factor $Q(T)$ and check for zeros in order to finish the decoding.

CHAPTER 5
CONCLUSION

We have provided a decoding algorithm for multipoint codes that takes advantage of existing list decoding algorithms. Additionally, we have improved the bound on the number of codewords output by the algorithm. It would be worthwhile to obtain a better estimate on the number of codewords output in the list decoding algorithm, because the decoding algorithm for multipoint codes requires us to check the output list of codewords in the supercode. However, we may not necessarily be interested in the worst case scenario, but rather the average list size. In [27], McEliece shows that the average list size using a Guruswami-Sudan decoder on Reed-Solomon codes is near one, so it often results in unique decoding. A similar result in the AG code setting would be quite interesting.

We have shown that there exists a function so that a multipoint code C is isometric to a subcode of a one-point code. However, it is not clear how to find that function efficiently, and it is not obvious how to choose the best one. While using the greatest common divisor as in Section 4.3 provide an approach for handling multiple functions, we would like to better understand a single "best" function for the shift.

The algorithm presented in this dissertation was analyzed for unique decoding of multipoint AG codes. However, this method could be extended to yield a list decoding algorithm for multipoint codes. In order to do this properly, however, one would want to further investigate properties associated with list decoding such as decoding radius.

INDEX

K-rational point, 25

affine point, 25
AG code, 26
AG code (over a ring), 48
AG code, dual, 29
algebraic function field, 6
algebraic geometry code, 26
algebraic geometry code (over a ring), 48
alphabet, 4

code, 4
code, $[n, k, d]$, 5
code, m-point, 26
code, dimension, 4
code, dual, 6
code, minimum distance, 5
codeword, 4

degree (of a divisor), 11
discrete valuation, 9
divisor, 10, 11
divisor, canonical, 18
divisor, dimension, 15
divisor, pole, 11
divisor, principal, 11
divisor, zero, 11

evaluation map, 27

gap number, 19
generator matrix, 29
genus, 17
Guruswami-Sudan algorithm, 34, 39

Hamming sphere, 5
Hermitian code, 27
Hermitian code, dual, 30
Hermitian function field, 7, 9
Hermitian function field, divisors, 13, 14
Hermitian function field, gap set, 21
Hermitian function field, genus, 17
Hermitian function field, homogeneous polynomial, 24

inner product, 6

local parameter, 8

parity check matrix, 29
place, 8
point at infinity, 25
pole, 10
pole number, 19
projective curve, 24
projective plane, 23

Reed-Solomon code, 5
residue class field, 8
residue class map, 8
Riemann-Roch space, 14
Riemann-Roch Theorem, 18

Singleton Bound, 29
Sudan's algorithm, 32
support, 10
Suzuki code, 61
Suzuki curve, 61

valuation ring, 8

Weierstrass Gap Theorem, 20
weight, 5

zero, 10

BIBLIOGRAPHY

1. M. Armand. Improved list decoding of generalized Reed-Solomon and alternant codes over Galois rings. *IEEE Trans. Inform. Theory*, 51(2):728–733, 2005.

2. M. Armand. List decoding of generalized Reed-Solomon over commutative rings. *IEEE Trans. Inform. Theory*, 51(1):411–419, 2005.

3. A. Barg, E. Krouk, and H. C. van Tilborg. On the complexity of minimum distance decoding of long linear codes. *IEEE Trans. Inform. Theory*, 45(5):1392–1405, 1999.

4. K. Bartley. Decoding algorithms for algebraic geometric codes over rings. *Ph.D. Thesis, University of Nebraska*, 2006.

5. K. Bartley and J. Walker. Algebraic geometric codes over rings. *Advances in Algebraic Geometry Codes, Series on Coding Theory and Cryptography (World Scientific)*, March 2008.

6. P. Beelen. The order bound for general algebraic geometric codes. *Finite Fields Appl.*, 13(3):665–680, 2007.

7. P. Beelen and T. Høholdt. List decoding using syndromes. *Algebraic Geometry and its Appl.*, 5:315–331, 2008.

8. A. Calderbank, A. Hammons Jr., P. V. Kumar, N. Sloane, and P. Sole. A linear construction for certain Kerdock and Preparata codes. *Bull. Amer. Math. Soc. (N.S.)*, 29(2):218–222, 1993.

9. C. Carvalho, C. Munuera, E. da Silva, and F. Torres. Near orders and codes. *IEEE Trans. Inform. Theory*, 53(5):1919–1924, 2007.

10. C. Carvalho and F. Torres. On Goppa codes and Weierstrass gaps at several points. *Des. Codes Cryptogr.*, 35(2):211–225, 2005.

11. P. Elias. List decoding for noisy channels. *Res. Lab of Electron., MIT, Cambridge, MA, Tech. Rep.*, 335, 1957.

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

13. V. Guruswami and A. Rudra. Limits to list decoding Reed-Solomon codes. *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 602–609, 2005.

14. V. Guruswami and M. Sudan. On representations of algebraic-geometry codes. *IEEE Trans. Inform. Theory*, 45:1757–1767, 1999.

15. V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Trans. Inform. Theory*, 47(4):1610–1613, 2001.

16. J. Hansen and H. Stichtenoth. Group codes on certain algebraic curves with many rational points. *Appl. Alg. Eng. Comm. Comput.*, 1(1):67–77, 1990.

17. M. Homma and S.J. Kim. Toward the determination of the minimum distance of two-point codes on a Hermitian curve. *Des. Codes Cryptogr.*, 37(1):111–132, 2005.

18. M. Homma and S.J. Kim. The two-point codes on a Hermitian curve with designed minimum distance. *Des. Codes Cryptogr.*, 38(1):55–81, 2006.

19. M. Homma and S.J. Kim. The two-point codes with the designed minimum distance on a Hermitian curve in even characteristic. *Des. Codes Cryptogr.*, 39(3):375–386, 2006.

20. A. Hammons Jr., P. V. Kumar, A. Calderbank, N. Sloane, and P. Sole. The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes. *IEEE Trans. Inform. Theory*, 40(2):301–319, 1994.

21. S. Ling and C. P. Xing. *Coding Theory: A First Course.* Cambridge University Press, 2004.

22. H. Maharaj, G. L. Matthews, and G. Pirsic. Riemann-Roch spaces of the Hermitian function field with applications to algebraic geometry codes and low-discrepancy sequences. *J. Pure Appl. Algebra*, 195(3):261–280, 2005.

23. G. L. Matthews. Weierstrass pairs and minimum distance of Goppa codes. *Des. Codes Cryptogr.*, 22:107–121, 2001.

24. G. L. Matthews. Codes from the Suzuki function field. *IEEE Trans. Inform. Theory*, 50(12):3298–3302, 2004.

25. G. L. Matthews. Weierstrass semigroups and codes from a quotient of the Hermitian curve. *Des. Codes Cryptogr.*, 37(3):473–492, 2005.

26. G. L. Matthews and T. W. Michel. One-point codes using places of higher degree. *IEEE Trans. Inform. Theory*, 51(4):1590–1593, 2005.

27. R. J. McEliece. On the average list size for the Guruswami-Sudan decoder. *Preprint.*

28. H. O'Keefe and P. Fitzpatrick. Gröbner basis approach to list decoding of algebraic geometry codes. *Appl. Algebra Engrg. Comm. Comput.*, 18(5):445–466, 2007.

29. H. O'Keeffe and P. Fitzpatrick. Gröbner basis solutions of constrained interpolation problems. *Linear Algebra Appl.*, 351/352:533–551, 2002.

30. S. Sakata and M. Fujisawa. Fast decoding of two-point AG codes. *Preprint.*

31. S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Høholdt. Fast decoding of AG-codes up to the designed minimum distance. *IEEE Trans. on Inform. Theory*, 41:1672–1677, 1995.

32. M. A. Shokrollahi and H. Wasserman. List decoding of algebraic-geometric codes. *IEEE Trans. Inform. Theory*, 45:432–437, 1990.

33. H. Stichtenoth. *Algebraic Function Fields and Codes.* Universitext. Springer-Verlag, 1993.

34. M. Sudan. Decoding of Reed-Solomon codes beyond the error correction bound. *Journal of Complexity*, 13:180–193, 1997.

35. J. Walker. Algebraic geometric codes over rings. *J. Pure Appl. Algebra*, 144(1):91–110, 1999.

36. J. Walker. *Codes and Curves.* Student Mathematical Library (**7**). IAS/Park City Mathematical Subseries. AMS, Providence, RI; IAS, Princeton, NJ, 2000.

37. J. Walker. *Private communication*, 2006.

38. M. Wang. Parameter choices on Guruswami-Sudan algorithm for polynomial reconstruction. *Finite Fields Appl.*, 13(4):877–886, 2007.

39. J. M. Wozencraft. List decoding. *Res. Lab of Electron., MIT, Cambridge, MA, Tech. Rep.*, 48:90–95, 1958.

40. C. P. Xing and H. Chen. Improvements on parameters of one-point AG codes from Hermitian codes. *IEEE Trans. Inform. Theory*, 48(2):535–537, 2002.

41. L. Xu. Improvement on parameters of Goppa geometry codes from maximal curves using the Vlădut-Xing method. *IEEE Trans. Inform. Theory*, 51(6):2207–2210, 2005.

42. K. Yang and P. V. Kumar. On the true minimum distances of Hermitian codes. *Lecture Notes in Mathematics*, pages 99–107, 1992.