NTNU
Norwegian University of
Science and Technology

Decoding of Algebraic Geometry Codes

Anna Aarstrand Slaatsveen

Master of Science in Physics and Mathematics
Submission date: June 2011
Supervisor: Kristian Gjøsteen, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

”Shannon is one of the great men of the century. Without him, none of the things we know today would exist.
The whole digital revolution started with him”

- Neil Sloane

Abstract

Codes derived from algebraic curves are called algebraic geometry (AG) codes. They provide a way to correct errors which occur during transmission of information. This paper will concentrate on the decoding of algebraic geometry codes, in other words, how to find errors.

We begin with a brief overview of some classical result in algebra as well as the definition of algebraic geometry codes. Then the theory of cyclic codes and BCH codes will be presented. We discuss the problem of finding the shortest linear feedback shift register (LFSR) which generates a given finite sequence. A decoding algorithm for BCH codes is the Berlekamp-Massey algorithm. This algorithm has complexity $\mathcal{O}(n^2)$ and provides a general solution to the problem of finding the shortest LFSR that generates a given sequence (which usually has running time $\mathcal{O}(n^3)$). This algorithm may also be used for AG codes.

Further we proceed with algorithms for decoding AG codes. The first algorithm for decoding algebraic geometry codes which we discuss is the so called basic decoding algorithm. This algorithm depends on the choice of a suitable divisor F. By creating a linear system of equation from the bases of spaces with prescribed zeroes and allowed poles we can find an error-locator function which contains all the error positions among its zeros. We find that this algorithm can correct up to $\lfloor (d^* - 1 - g)/2 \rfloor$ errors and have a running time of $\mathcal{O}(n^3)$. From this algorithm two other algorithms which improve on the error correcting capability are developed.

The first algorithm developed from the basic algorithm is the modified algorithm. This algorithm depends on a restriction on the divisors which are used to build the code and an increasing sequence of divisors $F_1 \leq \ldots \leq F_s$. This gives rise to an algorithm which can correct up to $\lfloor (d^* - 1)/2 - S(H) \rfloor$ errors and have a complexity of $\mathcal{O}(n^4)$. The correction rate of this algorithm is larger than the rate for the basic algorithm but it runs slower.

The extended modified algorithm is created by the use of what we refer to as special divisors. We choose the divisors in the sequence of the modified algorithm to have certain properties so that the algorithm runs faster. When $\sigma(\varepsilon)$ is the Clifford's defect of a set ε of special divisor, the extended modified algorithm corrects up to $\lfloor (d^* - 1)/2 - \sigma(\varepsilon) \rfloor$ which is an improvement from the basic algorithm. The running time of the algorithm is $\mathcal{O}(n^3)$.

The last algorithm we present is the Sudan-Guruswami list decoding algorithm. This algorithm searches for all possible code words within a certain distance from the received word. We show that AG codes are (e, b)-decodable and that the algorithm in most cases has a a higher correction rate than the other algorithms presented here.

Acknowledgements

I would like to thank my supervisor Kristian Gjøsteen for all the support and valuable advice during my studies in code theory and the preparation of this report.

Further I am grateful to Dr. Richard Mollin at the University of Calgary who during my year abroad inspired me to choose code theory as my main subject and introduced me to the intruding field of discrete mathematics and number theory.

I would also like to thank my mathematics teacher in Ungdomsskolen (8th-10th grade), Eric Bisgaard Lien for inspiring me and making me promise to continue studying mathematics. This promise has been a real motivator during my University years and in particular when working on my Masters Thesis.

I am indebted to my fellow students at NTNU and University of Calgary for making my five years at university enjoyable an manageable. The support and company of friends have been very important for reaching my goals.

Last but not least I am grateful to my dad for taking the time and effort to read through my paper.

Contents

1 Introduction

The theory of error-correcting codes concerns the tools used to transmit information safely over an unreliable channel. In other words coding theory is primarily concerned with dealing with errors created by noise. An example from every day use is communication by cellular phones. When using a cellular phone, signals are transmitted using the air as a communication channel. Because of electromagnetic interference the receiving end pick up noise as well as the message. In many areas of natural communication, error correction is used. Technological communication would be difficult or even impossible without being able to correct errors created by noise.

Claude Shannon published a paper which changed the way we look at communication for ever and because of this, he is known as the father of information theory. He added the effect of noise into the equation of communication. He constructed a way to encode data so that it could be decoded to a specified degree of accuracy after transmission.

The following figure shows the steps of a coding-decoding cycle of an error-correcting code.

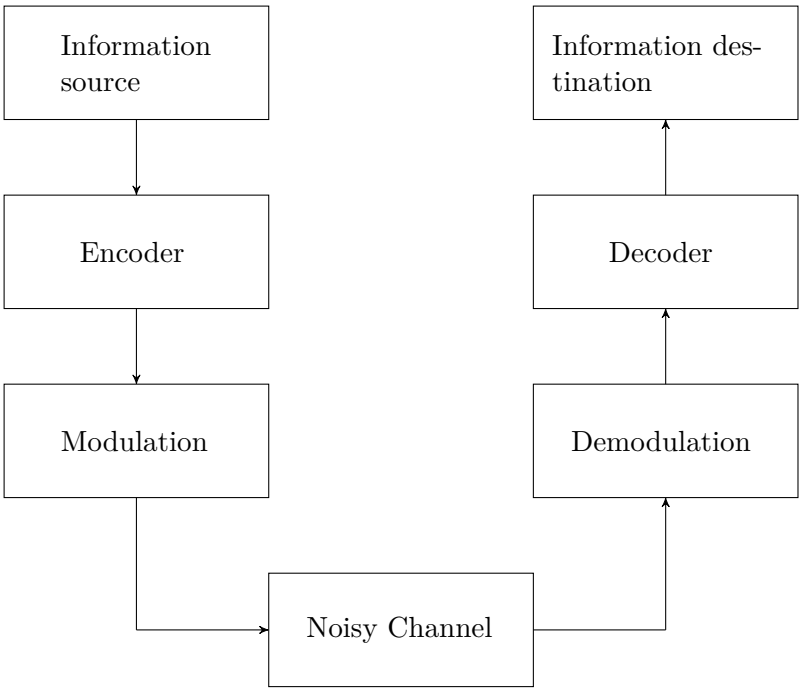

Figure 1: A canonical digital communication system

The information source transmits a block of k bits thorough a channel of some sort. This k bit string is called the information word. Some codes are built on information streams of

arbitrary length (infinite length). But in this paper we will restrict to block codes defined for some finite length n. To send more information than n bits we split the message into several n-bit blocks and send them separately. Since each block is assumed independent with respect to content and effect of noise, we only need to look at the case of transmitting one block of information. Thus when taking about codes we actually mean block codes. The information may be anything from a picture to data on a disk to sound. When transmitting the desired information, some of the bits may be altered by noise and the information which reach the destination may be corrupted. The difficult part of error-correction codes is to retrieve the original information even though it has been altered by noise.

To handle this problem we send more information than the original message. This is done by adding redundancy to the information word. This expanded bit string is called a code word. By adding redundancy, we make it possible to recover the original message even though error have occurred during transmission. If we have an easy channel, we may apply what is called repetition codes. This is as simple as for each bit we want to send, we repeat it several times. Thus, if every bit of a message is repeated seven times the receiver checks every group of seven bits to decode the message. If it contains more 0's than 1's, it is assumed that a 0 was sent, and vice versa. (Of course this is a fairly simple way to add redundancy and it will usually be as simple.)

Encoding is the process of adding redundancy to the information word. This is a function denoted by enc and takes an k-bit string to an n-bit string. If Σ is the alphabet the encoder function is as follows:

$$enc : \Sigma^k \longrightarrow \Sigma^n, \quad \text{where} \quad n > k$$

A code word is the word output from the encoder and is the information which is transmitted over the unreliable channel. Our code $\mathcal{C}$ is the set of such code words that is possible to transmit. Hence

$$\mathcal{C} = \{y : y = enc(x), x \in \Sigma^n\}$$

The length of the code is defined to be n and each code word in the code is an n-tuple with entries from the alphabet Σ. The dimension of the code is referred to as k. In this text, only linear codes are discussed and all codes are thus k-dimensional vector spaces.

The channel is the communication medium over which the transmission occurs. We denote the received information by y', i. e. the output from the unreliable channel. Hence y' is the code word y effected by the noise in the communication medium. Decoding denotes the process used on the receiving end to recover the original message. The decoder, denoted dec, is the algorithm that the receiver use to recover the original information from y'. Hence $x = dec(y')$ and takes a n-bit string to a k-bit string.

2 Algebraic Curves

In this section the theory behind algebraic curves are presented. This includes an introduction to affine and projective spaces, hypersurfaces, algebraic sets and function fields. Two examples of families of curves which commonly is used to build algebraic geometry codes will be given. For both the families of curves a group law on points of the curve will be derived.

2.1 Algebraic Curves

Let $\mathbb{A}^n$ denote the n-dimensional affine space and $\mathbb{A}^n(\mathbb{F})$ the affine n-space over the field $\mathbb{F}$. The affine space over a field $\mathbb{F}$ is the cartesian product of $\mathbb{F}$, n times. Hence elements or points in the affine space are n-tuples of the form $x = (x_1, x_2, \ldots x_n)$ where x_i is in $\mathbb{F}$ for all i. We say that $x_1, x_2, \ldots x_n$ are the coordinates of the point x.

In the affine plane $\mathbb{A}^2$ most lines will intersect at a point. The only exception is parallel lines which do not have a intersection point. In the projective plane, there will always exists a intersection point between any two lines even when they are parallel. The points in which parallel lines intersect are refereed to as the points at infinity.These intersection points may be computed in the same way as other intersection points.

In the affine plane we identify a point (x, y) in $\mathbb{A}^2$ with the point $(x, y, 1) \in \mathbb{A}^3$. Every point $(x, y, 1)$ in $\mathbb{A}^3$ determines a line through $(0, 0, 0)$ and $(x, y, 1)$. All lines thorough $(0, 0, 0)$ in $\mathbb{A}^3$ will satisfy this except the lines in the z-plane (since $z = 0$). We say that the points at infinity in $\mathbb{A}^3$ corresponds to the lines in the z-plane which intersect $(0, 0, 0)$. This can be adapted to fit in an affine n-space.

Definition 1. A *projective n-space* over a field $\mathbb{F}$ is defined to be all the lines through $(0, 0, \ldots, 0)$ in $\mathbb{A}^{n+1}(\mathbb{F})$. We denote the projective n-space by $\mathbb{P}^n$.

Hence, every point $x = (x_1, \ldots, x_{n+1})$ in the affine $(n + 1)$-space, determine an element in the projective n-space, namely the line through x. If two points x and y have the property that for all coordinates x_i and y_i, $x_i = \lambda y_i$, where λ is in the field $\mathbb{F}$, we say that the two points determine the same line and x and y are equivalent (i. e. the points are in the same equivalence class). By this definition we say that all the lines $\{(\lambda x_1, \ldots, \lambda x_{n+1}) \mid \lambda \in k\}$ which are determined by $x = (x_1, x_2, \ldots, x_{n+1}) \neq (0, 0, \ldots, 0)$ are called the points of $\mathbb{P}^n$. Thus the projective n-space is the equivalence classes of points in $\mathbb{A}^{n+1} \backslash (0, 0, \ldots, 0)$. We denote the equivalent class generated by x by $[x_1 : x_2 : \ldots : x_n]$ and write

$$\mathbb{P}^n = \{[a_1 : a_2 : \ldots : a_{n+1}] \mid a_i \in \mathbb{F} \text{ not all of the } a_i\text{'s are zero } \}$$

Points of the form $[a_1 : a_2 : \ldots : a_n : 0]$ are called the points at infinity.

An equation like $Y = X^2$ is meaningful in the affine plane. But this is not the case for the projective plane. An equality can only hold if it is unaffected when all the coordinates of a point are multiplied by the same non-zero constant. Thus, for the projective plane one

needs homogeneous equations, equations where each term have the same degree. If we do an homogenization of $Y = X^2$ we obtain $YZ = X^2$ which hold in the projective plane.

Example 1. Let χ be a curve in the affine plane $\mathbb{A}^2$ given by the equation $Y^2 = X^3+9X+4$. The homogenization of this curve is $Y^2Z = X^3+9XZ^2+4Z^3$ and defines the corresponding projective curve in the projective plane. Note that we can retrieve the affine curve from the projective curve by choosing Z to be equal to 1, i. e. evaluate the homogenous curve in $(X, Y, 1)$.

Now, let F be a polynomial in $\mathbb{F}[X_1, X_2, \ldots, X_n]$. The set of zeros on F is called a hypersurface of F. This set describes the zeros of the polynomial and is therefore also called the set of zeros of F. We denote the hypersurface of F by

$$V(F) = \{P = (a_1, a_2, \ldots, a_n) \mid F(P) = 0, a_i \in \mathbb{F}, i = 1, 2, \ldots, n\}$$

Let $S = \{F_1, F_2, \ldots, F_r\}$ be a set of polynomials where F_i is in $\mathbb{F}[x_1, x_2, \ldots x_n]$ for $i = 1, 2, \ldots, r$. In a similar manner the hypersurface of the set S is defined as

$$V(S) = \{P \in \mathbb{A}^n(\mathbb{F}) \mid F(P) = 0 \ \forall \ F \in S\}$$

Algebraic sets or affine algebraic sets in $\mathbb{A}^n(\mathbb{F})$ are important in algebra. They emerge from the concept of hypersurfaces. A set X in the affine n-space is an affine algebraic set if it is equal to the set of zeros of some set S in $\mathbb{A}^n(\mathbb{F})$. Hence, a set in the affine space is algebraic if it is equal to a hypersurface of any set in $\mathbb{F}[x_1, \ldots, x_n]$. The affine n-space is itself algebraic since we can write $\mathbb{A}^n = V(0)$. Another example is the empty set ($V(1) = \emptyset$). By writing $(p_1, p_2, \ldots, p_n) = V(x_1 - p_1, x_2 - p_2, \ldots, x_n - p_n)$ we clearly see that any point in $\mathbb{A}^n$ is an algebraic set. For algebraic sets the following hold.

Proposition 1. *Let X and Y be algebraic sets in $\mathbb{A}^n(\mathbb{F})$. Then the union of X and Y is also an algebraic set. Furthermore, let $\{Z_\alpha\}$ be any collection of algebraic sets in $\mathbb{A}^n(\mathbb{F})$. The intersection of these sets is an algebraic set.*

An affine algebraic set A is reducible if A can be written as the union of two algebraic sets A_1 and A_2 neither equal to A.

Definition 2. An affine algebraic set A is *irreducible* if for any algebraic sets A_1 and A_2, if $A = A_1 \cup A_2$ then either $A_1 = A$ or $A_2 = A$.

Any affine algebraic set can be written as a union of a finite number of irreducible affine algebraic sets.

An irreducible algebraic set is called a variety. For a nonempty variety V we define its coordinate ring as

$$\Gamma(V) = \mathbb{F}[X_1, X_2, \ldots, X_n]/I(V)$$

Since $I(V)$ is a prime ideal an affine variety, $\Gamma(V)$ is a domain. For $\Gamma(V)$ we can form the function field of V by

$$K(V) = \{z \mid z = \frac{a}{b} \text{ for } a, b \in \Gamma(V) \text{ with } \}$$

This is the field of rational functions. The requirement that every element in $K(V)$ must be on the form a/b ensures that elements in the same equivalence class evaluates to the same value. The local ring $\mathcal{O}_P(V)$ is the set of all rational functions on V which are defined on P.

For a rational function f on V we say that f is defined at a point P if for some $a, b \in \Gamma(v)$, $f = \frac{a}{b}$ and $b(P) \neq 0$. If $a(P) = 0$, f has a zero at P. If f is not defined for a point P in V, we say that f has a pole at this point and P is a singular point. The set of points $P \in V$ where a rational function is not defined is called the pole set. Any rational function has the same number of zeros and poles when counted properly which means that we need to consider the multiplicity of the zeros and poles.

Example 2. Again we consider the equation $Y^2 = X^3 + 9X + 4$. Consider the rational function $1/y$. Since we can write

$$\frac{1}{y} = \frac{Z}{Y} \tag{2.1}$$

Since for $P_\infty = (0 : 1 : 0)$, $Z(P_\infty) = 0$ and $Y(P_\infty) = 1$, $1/y$ has a zero at P_∞.

In a similar manner, as for the affine space, a subset X in $\mathbb{P}^n(\mathbb{F})$ is a projective algebraic set if there exists a set of homogeneous polynomials S in $\mathbb{F}[X_1, X_2, \ldots, X_{n+1}]$ such that

$$X = V(S) = \{P \in \mathbb{P}^n(\mathbb{F}) \mid F(P) = 0 \ \forall F \in S\}$$

The same properties as for affine sets hold for projective algebraic sets. And in a similar manner we can define the function field of projective variety.

We define algebraic curves as

Definition 3. An *affine (projective) algebraic curve* χ is defined to be the zero set of a polynomial (homogenous polynomial) in $\mathbb{A}^2$ ($\mathbb{P}^2$), i. e.

$$\chi : \{f(X, Y) = 0\} \in \mathbb{A}^2 \qquad (\chi : \{f(X, Y, Z) = 0\} \in \mathbb{P}^2)$$

The $\mathbb{F}$-rational points of χ are the solutions of $F(X, Y) = 0$ ($F(X, Y, Z) = 0$) with $X, Y \in \mathbb{F}$. The set of $\mathbb{F}$-rational points is denoted $\chi(\mathbb{F})$.

Example 3. Consider the elliptic curve $\mathcal{E}$ given by $Y^2 = X^3 + 9X + 4 \pmod{13}$. By finding points so that the polynomial $f(X, Y, 1) = Y^2 - (X^3 + 9X + 4)$ evaluates to zero, we find that this curve has 13 rational points in $\mathbb{F}_{13}$ plus a point a infinity P_∞:

$P_\infty = (0 : 1 : 0)$	$P_4 = (1 : 12 : 1)$	$P_8 = (8 : 4 : 1)$	$P_{12} = (11 : 11 : 1)$
$P_1 = (0 : 2 : 1)$	$P_5 = (2 : 2 : 1)$	$P_9 = (6 : 12 : 1)$	$P_{13} = (4 : 0 : 1)$
$P_2 = (0 : 11 : 1)$	$P_6 = (2 : 11 : 1)$	$P_{10} = (8 : 9 : 1)$	
$P_3 = (1 : 1 : 1)$	$P_7 = (6 : 1 : 1)$	$P_{11} = (11 : 2 : 1)$	

To help count multiplicities of points at a rational curve we introduce the order function. For a rational function f, the order function $\mathrm{ord}_P(f)$ gives us the multiplicity of f at the point P. For an element $x = \frac{f'}{g'}$ in $K(\chi) = \{\frac{f}{g} \mid f, g \in \chi\}$, P in χ and a non-zero $c \in \mathbb{F}$ the order function has the following properties:

$$\begin{aligned} \mathrm{ord}_P(x) &= \mathrm{ord}_P(f') - \mathrm{ord}_P(g') \\ \mathrm{ord}_P(f' + g') &\geq \min\{\mathrm{ord}_P(f'), \mathrm{ord}_P(g')\} \\ \mathrm{ord}_P(c) &= 0 \end{aligned}$$

The order function can also determine the poles and zeros of a function. Let x and P be as above. Then, if the order of x at a point P is $\mathrm{ord}_P(x) > 0$ then f has a zero at P. On the other hand, for x to have a pole at P, $\mathrm{ord}_P(x)$ is negative. If $\mathrm{ord}_P(x) \geq 0$ then x is said to be regular or defined at P, and we can evaluate $x(P)$. Otherwise x has a pole at P and we write $x(P) = \infty$.

Example 4. We look at the curve $\mathcal{E}$ from Example 3. Consider the functions x and y in the function field of E. These functions can be written

$$x = \frac{X}{Z} \qquad \text{and} \qquad y = \frac{Y}{Z} \tag{2.2}$$

If we consider the points $P_1 = (0 : 2 : 1)$, $P_2 = (0 : 11 : 1)$, $P_{13} = (4 : 0 : 1)$ and $P_\infty = (0 : 1 : 0)$ we obtain that

$$\begin{aligned} &\mathrm{ord}_{P_1}(x) = 1 \qquad \mathrm{ord}_{P_2}(x) = 1 \qquad \mathrm{ord}_{P_\infty}(x) = -2 \\ &\mathrm{ord}_{P_{13}}(y) = 3 \qquad \mathrm{ord}_{P_\infty}(y) = -3 \end{aligned}$$

This implies that x has a zero of order 1 at P_1 and P_2 and a pole of order 2 at P_∞ while y has a zero of order 3 at P_{13} and a pole of order 3 at P_∞.

2.2 Explicit Families of Curves

Error-correcting codes derived from algebraic curves are what we refer two as algebraic geometry codes. Here two families of curves which are commonly used to build algebraic geometric codes are defined. For both these families of curves we derive the group law on their points.

2.2.1 Elliptic Curves

The first family of curves we will discuss are the elliptic curves. Elliptic curves are smooth curves with a specific base point. These curves are important for algebraic geometry codes because of the group law that exists on their points.

An elliptic curve denoted $\mathcal{E}$ over a finite field $\mathbb{F}$ with characteristic that are neither 2 nor 3 is defined as

$$y^2 = x^3 + ax^2 + bx + c$$

with coefficients a, b and c in $\mathbb{F}$. The discriminant of $\mathcal{E}$ is $\Delta = -16(4a^3 + 27b^2)$. For a smooth curve the discriminant must be different from zero which means that a and b must satisfy the condition that $\Delta \neq 0$. By applying the Riemann-Roch theorem, the equation for an elliptic curve with coefficients in any field can be represented as

$$\mathcal{E} : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.3}$$

This is what we call the Wierstrass representation for elliptic curves. This is the non-homogeneous representation of the Wierstrass equation. As well as containing all the points satisfying the Wierstrass equation, elliptic curves also contain an extra point $P_\infty = (0 : 1 : 0)$ at infinity. When $\mathcal{E}$ is defined over a finite field $\mathbb{F}$, the coefficients $a_1, \ldots, a_6$ are in $\mathbb{F}$. The homogeneous representation of the Weierstrass representation is

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

This representation has the same base point as the non-homogeneous with coefficients $a_1, \ldots, a_6$ in $\bar{\mathbb{F}}$.

For a curve $\mathcal{E} \subset \mathbb{P}^2$ given by the Wierstrass representation, $\mathcal{E}$ consist of all points which satisfies $F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0$ and the point $P_\infty = (0 : 1 : 0)$ at infinity. The set of points on an elliptic curve $\mathcal{E}$ over the field $\mathbb{F}$ is

$$\mathcal{E}(\mathbb{F}) = \{(x, y, 1) \mid F(x, y) = 0\} \cup \{(0 : 1 : 0)\}$$

Any line L in the projective plane, L intersects $\mathcal{E}$ in exactly three points. If L is a tangent to a point on the curve, this point counts twice. The fact that, when counting multiplicities, the intersection number between $\mathcal{E}$ and L is exactly 3 is a special case of Bézout's theorem (see [1]).

When we are familiarized with the general equation for elliptic curves, an addition law for points on the curve can be found. The decomposition law for elements in $\mathcal{E}$ is denoted $\oplus$ and defined as follows:

Definition 4. Let two points P and Q on an elliptic curve $\mathcal{E}$ lie on the line L. Denote the third point in the intersection between L and $\mathcal{E}$ by R. Let the line L' go through R and P_∞. The third intersection point between L' and $\mathcal{E}$ is then defined to be $P \oplus Q$.

The operator $\oplus$ have the following properties:

Proposition 2. *Let $\mathcal{E} \subset \mathbb{P}^2$ be an elliptic curve defined by the Weierstrass formula over a finite field $\mathbb{F}$. Then*

1. *If a line $L \subset \mathbb{P}^2$ intersects $\mathcal{E}$ in P, Q, R (not necessarily distinct points), then*

$$(P \oplus Q) \oplus R = P_\infty$$

2. *There exist an element P_∞ so that for all points P on $\mathcal{E}$, $P \oplus P_\infty = P$.*

3. *$P \oplus R = R \oplus P$ for all points $P, R \in \mathcal{E}$.*

4. *For all points P on $\mathcal{E}$ there is another point denoted $\ominus P$ so that*

$$P \oplus (\ominus P) = 0 \tag{2.4}$$

5. *Let P, Q, R be points on $\mathcal{E}$. Then*

$$(P \oplus Q) \oplus R = P \oplus (Q \oplus R) \tag{2.5}$$

For a proof see [25]. By checking the axioms that define an abelian group, it is easily seen that the points on $\mathcal{E}$ is an abelian group with P_∞ as the identity. As long as it is clear that we use the group operations $\oplus$ and $\ominus$ we use $+$ and $-$ respectively for simplicity.

To make it easier to use points on an elliptic curve to construct algebraic geometry codes, we develop equations for the operations in Proposition 2. First a formula for calculating the inverse of a point is found. Let P_0 be a point on an elliptic curve $\mathcal{E}$ given by the Weierstrass equation:

$$F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0 \tag{2.6}$$

Let $P_0 = (x_0, y_0)$ be a point in $\mathcal{E}(\mathbb{F})$. This means that $F(x_0, y_0) = 0$. Let L be the line which intersects in P_0 and P_∞. This line must intersect E in a third point which we denote by R. By part 1) and 2) of Proposition 2 we have that

$$P_\infty = (P_0 + P_\infty) + R = P_0 + R$$

Thus R must be equal to $-P_0$ by (2.4). This implies that $-P_0$ also lie on L. Clearly L must be given by $L : x - x_0 = 0$. By substituting $x = x_0$ into (2.6) we obtain

$$y^2 = (x_0^3 + a_2x_0^2 + a_4x_0 + a_6) - (a_1x_0 + a_3)y \tag{2.7}$$

Clearly y_0 is a solution to this equation since $P_0 = (x_0, y_0)$ is a point on $\mathcal{E}$. Since $-P_0$ is on L, the x-coordinate of $-P_0$ must be x_0. Thus another solution of the equation above is y_0', where y_0' is the y-coordinate of $-P_0$. To determine y_0' we write

$$F(x_0, y) = c(y - y_0)(y - y_0')$$

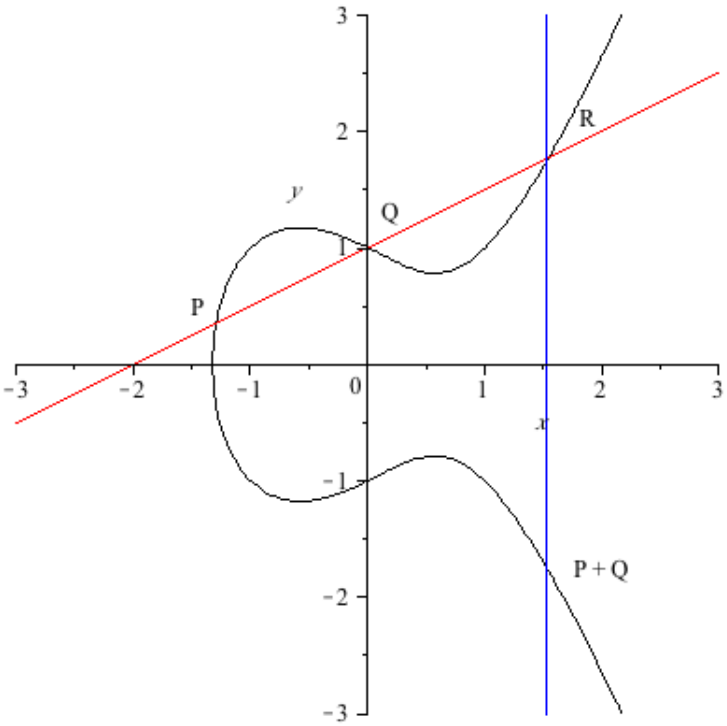

Figure 2: Addition of two distinct points on the elliptic curve $\mathcal{E} : \{y^2 = x^3 - x + 1\} \bigcup \{P_\infty\}$ over the reals. The points P, Q and R lie on the same line which mean that they add up to P_∞. We see that $-R$ is equal to $P + Q$.

The constant c must be equal to 1. Hence we have $F(x_0, y) = y^2 - y_0 y_0' + y_0'^2$. By comparing this result with (2.7), we obtain that y_0' is equal to $-(y_0 + a_1 x_0 + a_3)$. Hence

$$-P_0 = -(x_0, y_0) = (x_0, -(y_0 + a_1 x_0 + a_3)) \tag{2.8}$$

Next, we want to calculate an equation for the addition of two points, say $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ both on $\mathcal{E}$. The case where $x_1 = x_2$ and $P_1 \neq P_2$ is taken care of above, and the relationship between the y-coordinate of P_1 and P_2 is $y_1 = -(y_2 + a_1 x_2 + a_3)$. In this case $P_1 + P_2$ must be the point at infinity.

If the x-coordinate of P_1 and P_2 are different, the line which intersects both P_1 and P_2 has the form $L : y = \lambda x + \nu$ with slope $\lambda = (y_2 - y_1)/(x_2 - x_1)$. By substituting λ, x_1 and x_2 into the equation for L, the following expression for ν is obtained:

$$\nu = y - \frac{y_2 - y_1}{x_2 - x_1} x = \frac{y_1(x_2 - x_1) - (y_2 - y_1)x_1}{x_2 - x_1} = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}$$

What happens when $x_1 = x_2$ but $P_2 \neq -P_1$? The only possible answer is that y_1 and y_2 must be equal, thus P_1 is equal to P_2. In this case L is the tangent line to $\mathcal{E}$ at the point P_1 with multiplicity two. When using the standard method for calculation the slope of a tangent to a given curve at a certain point, the following is obtained

$$\lambda = \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}$$

Thus ν must be

$$\begin{aligned}\nu = y_1 - \lambda x_1 &= \frac{2y_1^2 + a_1x_1y_1 + a_3y_1 - (3x_1^3 + 2a_2x_1^2a_4x_1 - a_1y_1x_1)}{2y_1 + a_1x_1 + a_3} \\ &= \frac{(2y_1^2 + a_1x_1y_1 + 2a_3y_1 - 2x_1^3 - 2a_2x_1^2 - 2a_2x_1^2) - (x_1^3 + a_3y_1 + a_4x_1)}{2y_1 + a_1x_1 + a_3}\end{aligned}$$

By realizing that $2a_6 = 2y_1^2 + a_1x_1y_1 + 2a_3y_1 - 2x_1^3 - 2a_2x_1^2 - 2a_2x_1^2$ from $F(x_1, y_1) = 0$, the following expression for the intersection between L and the y-axis forms

$$\nu = \frac{2a_6 - x_1^3 + a_4x_1 - a_3y_1}{2y_1 + a_1x_1 + a_3}$$

By substitution for $y = \lambda x + \nu$ in the polynomial $F(x, y) = 0$ we obtain

$$F(x, \lambda x + \nu) = -x^3 + (\lambda^2 + a_1\lambda - a_2)x^2 + (2\lambda\nu + a_3\lambda - a_4)x + (a_3\nu + a_6) \tag{2.9}$$

The three roots of this third degree polynomial must be x_1, x_2 and x_3 where $P_3 = (x_3, y_3)$ is the third point in the intersection between L and $\mathcal{E}$. Thus another representation of $F(x, \lambda x + \nu)$ is

$$\begin{aligned}F(x, \lambda x + \nu) &= c(x - x_1)(x - x_2)(x - x_3) \\ &= c(x^3 - (x_1 + x_2 + x_3)x^2 + (x_1x_2 + x_1x_3 + x_2x_3)x - x_1x_2x_3)\end{aligned} \tag{2.10}$$

By comparing equation (2.9) and (2.10), the constant c must be equal to -1. Thus by equating the coefficients in front of the second degree term the expression for x_3 is

$$x_3 = \lambda^2 + a_1\lambda - (a_2 + x_1 + x_2)$$

By substituting for x_3 in the equation for L, we get an formula for y_3

$$y_3 = \lambda x_3 + \nu$$

Hence we have found the three points in the intersection between the elliptic curve E and L. Since P_1, P_2 and P_3 add up to P_∞ we obtain that

$$-P_3 = (x_3, -y_3) = P_1 + P_2$$

which completes the formulas for adding points on an elliptic curve. Next we will look at an example which uses some of the theory above.

Example 5. Consider the curve $Y^2 = X^3 + 9X + 4$ over $\mathbb{F}_{13}$. As seen in Example 11. This curve has 13 rational points plus a point at infinity. $P_5 = (2 : 2 : 1)$ and $P = (2 : 11 : 1)$ lie on the line $x = 2$. Note that $P_6 = -P_5$.

Now, let us see what happens when we add two points with different x-coordinates. Consider the points $P_3 = (x_3, y_3) = (1 : 1 : 1)$ and $P_9 = (x_9, y_9) = (6 : 12 : 1)$. Denote the line which intersect the elliptic curve in P_3 and P_9 by L. We have seen that this line has equation $y = \lambda x + \nu$ an we get

$$\lambda = \frac{y_9 - y_3}{x_9 - x_3} = \frac{12 - 1}{6 - 1} = 10 (\text{mod } 13)$$
$$\nu = \frac{y_3 x_9 - y_9 x_3}{x_9 - x_3} = 2 (\text{mod } 13)$$

Denote the third point in the intersection between L and the elliptic curve be $P' = (x', y')$. From the formulas calculated earlier

$$x' = \lambda^2 + a_1\lambda - (a_2 + x_9 + x_3) = 2 (\text{mod } 13)$$
$$y' = \lambda x' + \nu = 11 (\text{mod } 13)$$

Thus the third point on L is $P' = (2 : 11 : 1) = P_6$. Since $-P_6 = P_5$ we get

$$P_5 = (2 : 2 : 1) = P_3 + P_9$$

2.2.2 Hyperelliptic Curves

Another family of curves often used in coding theory are the hyperelliptic curves. These curves are nonsingular curves defined by

Definition 5. Let $\mathbb{F}$ be a field. Then for polynomials f and h in $\mathbb{F}[x]$ satisfying that f is monic, $\deg(f) = 2n + 1$ and $\deg(h) \leq n$ for some $n \in \mathbb{Z}$ greater than or equal to 2, a curve on the form

$$\mathcal{C} : y^2 + h(x)y = f(x) \tag{2.11}$$

is *hyperelliptic* over $\mathbb{F}$ if no point on the curve over the algebraic closure $\bar{\mathbb{F}}$ of $\mathbb{F}$ satisfies that both the partial derivatives $\frac{\partial}{\partial y}(y^2 + h(x)y - f(x)) = 2y + h$ and $\frac{\partial}{\partial x}(y^2 + h(x)y - f(x)) = h'(x)y - f'(x)$ are zero.

The conditions concerning the curve's partial derivatives ensure that the curve is smooth, i. e. the curve has no singularities. It is also a necessity that f has no multiple roots for the curve to be hyperelliptic.

For hyperelliptic curves in $\mathbb{P}^2(\mathbb{F})$ with coordinates $(X : Y : Z)$

$$\mathcal{C}(\mathbb{F}) = \{(x, y) \in F^2 \mid y^2 + h(x)y = f(x)\} \cup \{P_\infty\} \tag{2.12}$$

is the set of $\mathbb{F}$-rational points on the curve. Here P_∞ is equal to $(0:1:0)$. Consider a point $P_0 = (x_0, y_0) \in \mathcal{C}(\mathbb{F})$ different from the point at infinity and another point, $P = (x_0, -y_0 - h(x_0))$. Since

$$\begin{aligned}(-y_0 - h(x_0))^2 + h(x_0)(-y_0 - h(x_0)) &= y_0^2 - 2y_0h(x_0) + h(x_0)^2 - h(x_0)y_0 - h(x_0)^2 \\ &= y_0^2 - h(x_0)y_0\end{aligned}$$

P is also a point on the hyperelliptic curve $\mathcal{C}$. More specific, this point is called the opposite of P_0 and is denoted

$$-P_0 = (x_0, -y_0 - h(x_0))$$

If $P_0 = -P_0$ the point is called a Weierstrass point. Weierstrass points are points which are fixed under the hyperelliptic involution. Clearly P_0 and $-P_0$ lie on the line $x = x_0$.

By a change of variables and a restriction on the field $\mathbb{F}$, Defintion 5. can be simplified. We assume that the characteristic of the field $\mathbb{F}$ is not equal to 2 and consider the following change in variables

$$x \mapsto x \quad \text{and} \quad y \mapsto y - \frac{h(x)}{2}$$

The change in y makes sense as long as $\text{char}(F) \neq 2$. Under these changes

$$\begin{aligned}f(x) = (y - \frac{h(x)}{2})^2 + h(x)(y - \frac{h(x)}{2}) &= y^2 - yh(x) + \frac{h(x)^2}{4} + h(x)y - \frac{h(x)^2}{2} \\ &= y^2 - \frac{h(x)^2}{4}\end{aligned}$$

By definition $h(x)$ is a polynomial of degree less than or equal to $n = (\deg(f(x)) - 1)/2$. It follows that $\deg(h(x)^2) = 2\deg(h(x)) \leq 2n =$. Since $f(x)$ is a monic polynomial of degree $2n + 1$ for some integer n, greater than or equal to 2, $f(x) + h(x)^2/4$ must also be monic and of degree $2n+1$. As long as the field which the curve is defined over have characteristic different from 2, we can define an hyperelliptic curve as

$$\mathcal{C} : y^2 = f(x) \tag{2.13}$$

where $f(x)$ is a monic polynomial of degree $2n + 1$. As long as $f(x)$ has no multiple roots, $\mathcal{C}$ is non-singular.

To state the group law for hyperelliptic curves, we give a short overview of some important results without proof. To obtain a more thorough understanding on how the group law for this class of curves works see [26], in particular Chapter 4 and Chapter 14, or other similar readings.

For elliptic curves, the points on the curve and a point at infinity defines an abelian group. For the class of hyperelliptic curves, this is no longer true. Thus we need to define

a group on $\mathcal{C}$ so that we can apply the group law when adding elements on the curve. Let $\mathcal{C}$ be a hyperelliptic curve over $\mathbb{F}$ with equation $y^2 = f(x)$ where $f(x)$ is a polynomial of degree 5. By considering finite sums of points as group elements and perform addition coefficientwise as

$$(P + Q) \oplus (R + Q) = P + 2Q + R \quad \text{where} \quad P, R, Q \in C(\mathbb{F})$$

we have a solution to the problem. To prevent us from creating infinite groups and representation of group elements which are to long, the quotient group including all sums of points that lie on the curve is used. In other words, we build a group by taking the quotient of the group of sums of points which lie in the intersection between the hyperelliptic curve and a curve in the plane.

A cubic will intersects the hyperelliptic curve $\mathcal{C} : y^2 = f(x)$ in six points, say P_1, P_2, Q_1, Q_2, $-R_1$ and $-R_2$. These points add up to the zero element, P_∞, in the quotient group. In this case we realize that each point can be represented by at most two points that do not have the same x-coordinate and inverse y-coordinate.

Any m points on the curve gives rise to a polynomial of degree $m-1$. Besides these m points, there are $\max\{5, 2(m-1)\}-m$ other points of intersection between the hyperelliptic curve and the polynomial of degree $m-1$. Adding two elements are done in two steps. First the formal sum of points is reduced. Consider the group elements $P_1 + P_2$ and $Q_1 + Q_2$ consisting of 4 distinct points in the intersection between a cubic polynomial and hyperelliptic curve. By the formula above, there are another 2 points of intersection between these two curves. Denote these points by $-R_1$ and $-R_2$. We inflect these points to obtain the result of the addition

$$(P_1 + P_2) \oplus (Q_1 + Q_2) = (R_1 + R_2) \tag{2.14}$$

Hence, as long as we have $m > 2$ points, the inverse of the sum of points obtained by inflecting all points at the x-axis, contains fewer points. By repeating this process, a reduced group element is found containing at most 2 points.

The general result is; for hyperelliptic curves where $f(x)$ has degree equal to $2n+1$ for some $2 \leq n \in \mathbb{Z}$, each group element can be represented by at most n points. One might have to apply the reduction step several times before reaching the minimal representation of an element.

To generate the general algorithm for adding points on a hyperelliptic curve, the notion of divisors and divisor classes are needed. Let χ be an algebraic curve over a field $\mathbb{F}$. A divisor of the curve χ is a finite linear combination of points on the curve with integer coefficients.

Definition 6. A *divisor* on a curve χ is defined as a formal sum $D = \Sigma_{P \in \chi} n_P P$ where $n_P \in \mathbb{Z}$ and $n_P \neq 0$ for a finite number of points in χ. Here n_P is the *multiplicity* of the point P on the curve χ.

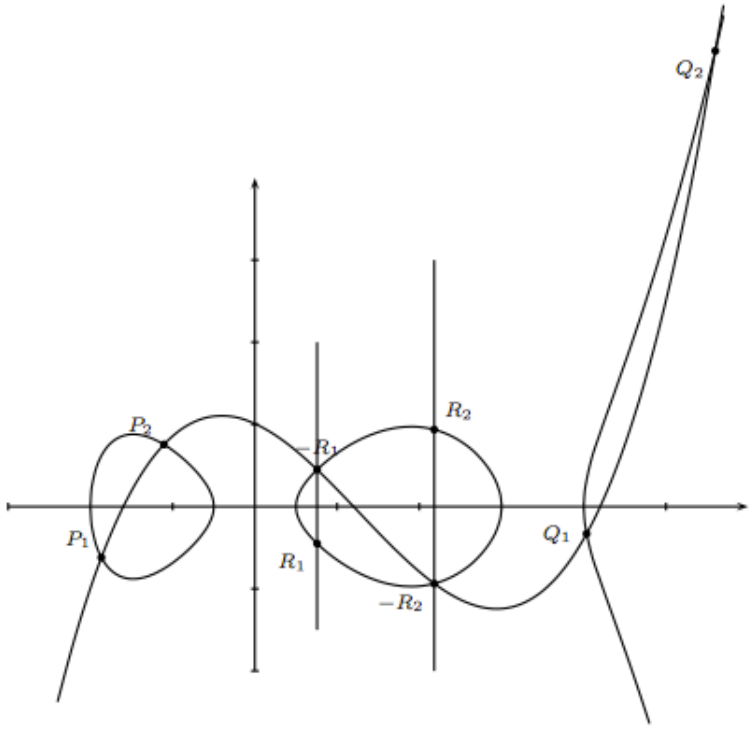

Figure 3: Example of the group law on a curve $\mathcal{C} : y^2 = f(x)$ over $\mathbb{R}$ with $\deg(f(x)) = 5$. Let $R_1 = (x_{R_1}, y_{R_1})$ and $-R_1 = (x_{R_1}, -y_{R_1})$. These points lie on the curve $x = x_{R_1}$ and are the opposite of each other. There are 6 points in the intersection between $\mathcal{C}$ and a cubic. These points add up to P_∞, i. e. $(P_1 + P_2) \oplus (Q_1 + Q_2) = R_1 + R_2$. Any line will intersect $\mathcal{C}$ in 5 points.

The degree of a divisor D as in the definition above is defined to be Σn_P. In other words the degree of a divisor is the sum of its coefficients. If all the non-zero coefficients in $D = \Sigma n_P P$ are positive, the divisor is said to be effective or integral ($n_P \geq 0$).

Two divisors A and B of a rational curve χ are said to be equivalent or in the same equivalence class if $A = B + \mathrm{div}(f)$ for some f in the function field of the curve. We write $A \equiv B$ when A and B are contained in the same equivalence class. Since the degree of the divisor of a rational function is zero, the degree of all divisors in an equivalence class are the same. A divisor D is only equivalent to zero if $D = \mathrm{div}(z)$ for some $z \in K(\chi)$.

Next we define what is called divisor class groups and semi-reduced divisors. To divisor class group of a hyperelliptic curve is defined by the divisors of degree zero which are the divisors on rational functions.

Definition 7. Let $\mathcal{C}$ be a hyperelliptic curve over $\mathbb{F}$. Then the *group of divisors of degree*

0 is given by

$$\mathrm{Div}_{\mathcal{C}}^{0} = \{D = \sum_{P \in \mathcal{C}} n_P P \mid n_P \in \mathbb{Z}, \sum_{P \in \mathcal{C}} n_P = 0 \text{ such that } \sigma(D) = D \forall \sigma \in G_{\mathbb{F}}\} \tag{2.15}$$

where $G_{\mathbb{F}}$ is the Galios group over $\mathbb{F}$. Thus $\mathrm{Div}_{\mathcal{C}}^{0}$ denotes the $\mathbb{F}$-rational divisors of $\mathcal{C}$.

Definition 8. The *divisor class group* of $\mathcal{C}$ is the quotient group of $\mathrm{Div}_{\mathcal{C}}^{0}$ by the group of principal divisors, that are divisors of degree zero resulting from rational functions. We denote the divisor class group by $\mathrm{Pic}_{\mathcal{C}}^{0}$.

Each divisor class can be uniquely defined by $\sum_{i=1}^{r} P_i - rP_\infty$ where P_i is an element of $\mathcal{C}(\mathbb{F})$ where none of the points in the sum are opposite of each other.

For curves of the form in Definition 5 the divisor class group is isomorphic to the group of K-rational points of the Jacobian $J_{\mathcal{C}}$ of $\mathcal{C}$. The Mumford's representation takes advantage of this isomorphism. Before deriving Mumford's Theorem we discuss the Jacobian $J_{\mathcal{C}}$ of an hyperelliptic curve $\mathcal{C}$.

A semi-reduced divisor is a divisor in which no two points are the opposite of each other. If such a divisor contains k points, the divisor is said to have weight k. If k is less than or equal to the integer n in Definition 5, the devisor is reduced. The Jacobian of a curve is the set of all reduced divisors on the curve. An addition operation can be defined on a reduced divisor, which makes $J_{\mathcal{C}}$ into a group (which is not possible on the curve). In Mumford's theorem a representation for reduced divisors is stated.

Theorem 1 (Mumford's Theorem)**.** *Let $\mathcal{C}$ be a hyperelliptic curve as in Definition 5. Then each non-trivial divisor class over $\mathbb{F}$ can be represented by a unique pair of polynomials $u(x)$ and $v(x)$ where $u, v \in \mathbb{F}[x]$ so that*

1. *u is monic.*
2. *$\deg(v) < \deg(u) \leq n$.*
3. *$u \mid v^2 + vh - f$.*

Let $D = \sum_{i=1}^{r} P_i - rP_\infty$, where $r \leq n$, $P_i \neq P_\infty$ and P_j is different from the opposite of P_i when $i \neq j$. Let $P_i = (x_i, y_i)$. Then the divisor class of D is represented by

$$u(x) = \prod_{i=1}^{r} (x - x_i)$$

If P_i occurs n_i times then

$$\left(\frac{d}{dx}\right)^{j} [v(x)^2 + v(x)h(x) - f(x)]_{|x=x_i} = 0 \quad \textit{for} \quad 0 \leq j < n_i$$

The theorem tells us that each divisor class can be represented by a reduced divisor. The second part of the theorem states that for each point $P_i = (x_i, y_i)$ of D, $u(x_i) = 0$. The last condition gives that $v(x_i) = y_i$. A divisor class which is represented by the polynomials $u(x)$ and $v(x)$ is denoted $[u(x), v(x)]$.

Most reduced divisors have weight n. By Mumford's Theorem, divisors containing only a single point $P = (x_P, y_P)$ is represented by $[u(x), v(x)] = (x - x_P, y_P)$. We also conclude that there is a unique divisor of weight 0 represented by $\mathcal{O} = [u(x), v(x)] = [1, 0]$. This is the neutral element (identity element) of the addition law which will be defined on the Jacobian. Scalar multiplication by an integer l is defined by

$$[l]D = \underbrace{D + D + \ldots D}_{l \text{ times}}$$

If $[l]D = \mathcal{O}$ then D is an l-torsion divisor. For curves with $\deg(f(x)) = 5$ and $\deg(h(x)) \leq 2$, divisors in $J_{\mathcal{C}}/\Theta$, where Θ is the reduced divisors of degree strictly less than 2, is of the form

$$D = [x^2 + u_1 x + u_2, v_0 x + v_1]$$

Cantor's algorithm uses the representing polynomials and polynomial arithmetic over a field $\mathbb{F}$ defined by Mumford's Theorem to define the group law for hyperelliptic curves. This is a sequence of composition and reduction, when taken into consideration that it holds for curves defined over any field.

Algorithm (Cantor's Addition Algorithm)**.** INPUT: $(\bar{D})_1 = [u_1, v_1]$, $(\bar{D})_2 = [u_2, v_1]$, which both are divisor classes of a hyperelliptic curve given by $\mathcal{C} : y^2 + h(x)y = f(x)$. OUTPUT: A unique reduced divisor such that $\bar{D} = (\bar{D})_1 \oplus (\bar{D})_2$.

1. $d_1 = \gcd(u_1, u_2)$, i. e. $d = e_1 u_1 + e_2 u_2$
2. $d = \gcd(d_1, v_1 + v_2 + h)$, i. e. $d = c_1 d_1 + c_2(v_1 + v_2 + h)$
3. $s_1 = c_1 e_1$, $s_2 = c_1 e_2$ and $s_3 = c_2$
4. $$u = \frac{u_1 u_2}{d^2} \quad \text{and} \quad x = \frac{s_1 u_1 v_2 + s_2 u_2 v_1 + s_3(v_1 v_2 + f)}{d} \mod u \tag{2.16}$$
5. $$u' = \frac{f - vh - v^2}{u} \quad \text{and} \quad v' = (-h - v) \mod u' \tag{2.17}$$
6. Update $u = u'$ and $v = v'$

7. If $\deg(u) \leq n$ proceed to the next step. Otherwise return to step 4.

8. Make u monic and return $[u, v]$.

When applying this algorithm on elliptic curves, one see that it agrees with the addition law for elliptic curves.

3 The Riemann-Roch Theorem

The main theorem of this section is the Riemann-Roch theorem. This theorem is important for the computation of the dimension of the dimension of the space of functions with prescribed zeros and allowed poles. The discussion of divisors will be continued here and we define some important classes of divisors, among other the canonical divisors.

Recall that a divisor of a given curve is a formal sum of points on the curve. From Definition 6 and the definition of the order function a divisor D can also be written as $\sum_{P \in \chi} \mathrm{ord}_P P$. If z is an element of the function field of χ in n variables, then the zero divisor and the pole divisor of z is defined to be

$$(z)_0 = \Sigma_{\mathrm{ord}_P(z)>0} \mathrm{ord}_P(z)$$
$$(z)_\infty = \Sigma_{\mathrm{ord}_P(z)<0} \mathrm{ord}_P(z)$$

respectively . The divisor of z is defined as $\mathrm{div}(z) = (z)_0 - (z)_\infty$. As mentioned before a rational function z has the same number of poles as zeros. Thus we obtain that the degree of the zero divisor of z is equal to the degree of the pole divisor of z, i. e. $\deg((z)_0) - \deg((z)_\infty) = 0$. This implies that

$$\deg(\mathrm{div}(z)) = \deg((z)_0 - (z)_\infty) = 0$$

for a rational function z.

Example 6. Again we consider the elliptic curve $Y^2 = X^3 + 9X + 4 (\mathrm{mod}\ 13)$. As seen earlier, this curve has 13 rational points in $\mathbb{F}_{13}$, among others $P_1 = (0 : 2 : 1)$, $P_2 = (0 : 11 : 1)$ and $P_{13} = (4 : 0 : 1)$ plus a point at infinity, $P_\infty = (0 : 1 : 0)$.

The intersection divisor of the elliptic curve and the curve with equation $X = 0$ is $P_1 + P_2 + P_\infty$, with the curve $Y = 0$ being $3P_{13}$ and with the curve $Z = 0$ being $3P_\infty$. From Example 4, the zero and pole divisor of x and y is

$$(x)_0 = P_1 + P_1 \qquad (x)_\infty = 2P_\infty$$
$$(y)_0 = 3P_{13} \qquad (y)_\infty = 3P_\infty$$

Since x is the quotient between X and Z and y is the quotient between Y and Z we get the following divisors

$$\begin{array}{ll}
\mathrm{div}(x) = P_1 + P_2 - P_\infty & \mathrm{div}(y) = 3P_{13} - 3P_\infty \\
\mathrm{div}(\frac{1}{x}) = -\mathrm{div}(x) = P_\infty - P_1 - P_2 & \mathrm{div}(\frac{1}{y}) = -\mathrm{div}(y) = 3P_\infty - 3P_{13} \\
\mathrm{div}(x^2) = 2\mathrm{div}(x) = 2P_1 + 2P_2 - 4P_\infty & \mathrm{div}(xy) = P_1 + P_2 + 3P_{13} - 5P_\infty \\
\mathrm{div}(x^3) = 3P_1 + 3P_2 - 6P_\infty & \mathrm{div}(x^2y) = 2P_1 + 2P_3 + 3P_{13} - 7P_\infty \\
\mathrm{div}(x^4) = 4P_1 + 4P_2 - 8P_\infty & \mathrm{div}(xy^2) = P_1 + P_2 + 6P_{13} - 8P_\infty \\
\mathrm{div}(y^2) = 6P_{13} - 6P_\infty & \mathrm{div}(\frac{1}{x-4}) = -\mathrm{div}(x-4) = 2P_\infty - 2P_{13}
\end{array}$$

Let χ be a curve over a field $\mathbb{F}$ and K the corresponding function field. A special kind of divisors are the canonical divisors. These are important for the existence of the Riemann-Roch theorem, which we will state later. A canonical divisor is defined using derivations and differentials. All canonical divisors on a given curve are contained in the same equivalence class, which means that all canonical divisors on a given curve have the same degree. It can be derived that for a canonical divisor on an algebraic curve the degree is $2g-2$. Here g is an integer called the genus. This is a notion which will be defined later on in this section.

We want the definition of differentials and derivations on algebraic curves to be close to the definitions in analysis. To make this happen, a derivation d is defined as a map from a ring R to an R-module M by $d(xy) = xd(y) = yd(x)$ for all x and y in R, here $\mathbb{F} \subset R$. By definition d is a $\mathbb{F}$-linear map. The definition of a derivation d demands that for $a \in \mathbb{F}$ $d(a) = 0$ and

$$\begin{aligned}
d(z^n) &= nz^{n-1}d(z) \\
d\left(\frac{x}{y}\right) &= \frac{1}{y}d(x) + \frac{1}{x}\frac{-1}{y^2}d(y) = \frac{1}{y}\left(d(x) - \frac{x}{y}d(y)\right)
\end{aligned}$$

This agrees with the well known definitions from analysis.

Let d be a derivation as above. Pick a z in the function field of the curve such that for some f and g in the ring R, $z = f/g$. This implies that $f = zg$. The derivation of f is $d(zg) = zd(g) + g\tilde{d}(z)$. Hence $\tilde{d}(z) = \frac{1}{g}(d(f) - zd(g))$. It can be shown that $\tilde{d}$ is a well defined derivation.

For differentials of projective curves to behave like differentials from calculus we define a differential of a ring R to be of the form $\sum_i x_i dy_i$, where both x_i and y_i are elements of the ring and d is a derivation. Let F be a free R-module on the set $\{[x] \mid x \in R\}$ and N a submodule of F such that N is generated by the following elements where $[x]$ is a symbol for all elements x of the ring R.

(i) $\{[x+y] - [x] - [y] \mid x, y \in R\}$

(ii) $\{[\lambda x] - \lambda[x] \mid \lambda \in k, x \in R\}$

(iii) $\{[xy] - x[y] - y[x] \mid y, x \in R\}$

Let $F/N = \Omega_{\mathbb{F}}(R)$ and dx the image of $[x]$ in $\Omega_k(R)$ such that the mapping

$$d : R \longrightarrow \Omega_{\mathbb{F}}(R)$$

is defined by $x \mapsto dx$. This d is a derivation (i. e. a $\mathbb{F}$-linear map). Since $N \subset F$, $\Omega_{\mathbb{F}}(R)$ is also an R-module and it is called the module of differentials.

Let $\Omega = \Omega_{\mathbb{F}}(K)$ be the space of differentials of K over $\mathbb{F}$ (the image of d where $d : \Omega \longrightarrow R$). Pick an element $\omega \neq 0$ in the space of differentials. Then $\mathrm{div}(\omega) = \sum \mathrm{ord}_P(\omega)P$. To define the order function of ω at a place $P \in \chi$, take an uniformizing parameter $t \in \mathcal{O}_P(\chi)$ so that there exists an $f \in K$ which satisfy $\omega = fdt$. Then, define $\mathrm{ord}_P(\omega) = \mathrm{ord}_P(f) + \mathrm{ord}_P(dt) = \mathrm{ord}_P(f)$. Hence, $\mathrm{div}(\omega) = \sum_{P \in \chi} \mathrm{ord}_P(f)P$. Divisors of this form are the canonical divisors. In other words, when w is a rational divisors then $W = (w)$ are canonical.

When working with algebraic geometry codes it is essential to be able to find functions that only have poles at a certain set of points so the information about where things might go "wrong" is known. If $D = \Sigma_{P \in \chi} n_P P$ is a divisor of a projective curve χ, we want to find a rational function with poles of degree no worse than n_P. With this in mind, define

$$L(D) = \{f \in K \mid \mathrm{ord}_P(f) \geq -n_P \ \forall P \in \chi\}$$

This space is often called the Riemann-Roch space. Note that the constant functions only are contained in $L(D)$ when D is effective. For two equivalent divisors A and B the Riemann-Roch spaces $L(A)$ and $L(B)$ are isomorphic as vectorspaces.

The definition of the Riemann-Roch space is central in the main theorem of this section, the Riemann-Roch theorem. This theorem depends on knowledge about the dimension of the Riemann-Roch space. Denote the dimension of $L(D)$ by $l(D)$. To help determine the size of the basis of $L(D)$ we introduce Riemann's theorem which gives a lower bound for $l(D)$.

Theorem 2 (Riemann's)**.** *For all divisors D on an algebraic curve χ there exists a $g \in \mathbb{Z}$ such that $l(D) \geq \deg(D) + 1 - g$. The smallest such integer g is called the genus of χ.*

The boundary on $l(D)$ depends on an integer g which is called the genus of an algebraic curve. The genus is a topological invariant which is defined for each curve $F(x, y) = 0$. When we see the curve as a Riemann surface, the genus represents the number of holes in the curve. For lines, conics and singular cubics the genus is zero. While non-singular cubics have genus 1. To determine the genus of a non-singular curve the following proposition may be applied

Proposition 3. *For a non-singular plane curve of degree d the genus is*

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

Thus, for non-singular curves the genus increases with the degree of the defining polynomial.

Example 7. The elliptic curve $\mathcal{E} : Y^2 = X^3 + 9X + 4$ is nonsingular with defining polynomial of degree 3. Thus we can apply Proposition 3 and get that $g = ((3-1)(3-2))/2 = 1$. Hence $\mathcal{E}$ has genus 1.

When we know have defined the invariant g, we can adapt Definition 5 to depend on the genus.

Example 8. Hyperelliptic curves are nonsingular curves of genus greater than or equal to 2 defined as follows: For polynomials f and h in $F[x]$ satisfying that f is monic, $\deg(f) = 2g+1$ and $\deg(h) \leq g$ a curve

$$\mathcal{C} : y^2 + h(x)y = f(x)$$

is called a hyperelliptic curve of genus g over F if no point on the curve over the algebraic closure $\bar{F}$ of F satisfies both the partial derivatives $\frac{\partial}{\partial y}(y^2 + h(x)y - f(x)) = 2y + h = 0$ and $\frac{\partial}{\partial x}(y^2 + h(x)y - f(x)) = h'(x)y - f'(x) = 0$. Note that if $g = 1$, this also agree with the definition for elliptic curves.

All curves of genus 2 are hyperelliptic curves. These are the curves where $h(x)$ has degree at most 2 and $f(x)$ is monic and has degree 5. For curves with genus strictly larger than 2 we can not necessarily draw the conclusion that the curve is hyperelliptic.

The next theorem gives a formula for the dimension of the Riemann-Roch space of a divisor on a given curve.

Theorem 3 (Riemann-Roch)**.** *Let W be a canonical divisor on an algebraic curve χ with genus g. Then for any divisor D on the same curve, the following holds*

$$l(D) - l(W - D) = \deg(D) + 1 - g \tag{3.1}$$

This result was used to develop some of the new aspects in coding theory.

In the next example we will use some of the theory discussed above

Example 9. Let $\mathcal{E}$ be the elliptic curve from Example 3. This curve has genus 1 and 13 rational points in $\mathbb{F}_{13}$, among others $P_1 = (0:2:1)$, $P_2 = (0:11:1)$ and $P_{13} = (4:0:1)$, and a point at infinity, $P_\infty = (0:1:0)$. We look at the divisor $G = 8P_\infty$ and find the basis of the Riemann-Roch space $L(G)$. By definition the degree of G is 8. The Riemann theorem tells us that $l(D) = 8$.
Since functions in $L(G)$ must have a pole at P_∞ with order less than or equal to 8 we get the following basis:

$$L(G) = \{1, x, x^2, x^4, y, y^2, xy, x^2y\}$$

Note that the functions x^3 and xy^2 also are elements in $L(G)$. But since x^3 can be written as $y^2 - (9x + 4)$ and xy^2 can be written as $x^4 + 9x^2 + 4x$ they are linear combinations of other elements in $L(G)$ and hence not elements of the basis of $L(G)$.

Now consider the divisor $F = 3P_{13}$. Since the curve E has genus 1 Riemann's theorem tell us that $l(F)$ is 3. The vector space $L(F)$ contains functions which have a pole at P_{13} of order 1, 2 or 3. From Example 6 we thus have that a basis for this vector space is

$$L(F) = \left\{1, \frac{1}{y}, \frac{1}{x-4}\right\}$$

Note that 1 is an element of $L(G)$ and $L(F)$ since both G and F are integral divisors.

The third and last divisor we look at is $G - F = 8P_\infty - 3P_{13}$. The degree of this divisor is $8 - 3 = 5$. Again we apply the Riemann theorem and get that $l(G - F) = 5$. Hence we search for a basis of $L(G - F)$ with 5 elements. Since $G - F$ is not effective the constant functions are not in the basis of $L(G - F)$ since functions $f \in L(G - F)$ have to satisfy $\text{ord}_{P_\infty}(f) \geq -8$ and $\text{ord}_{P_{13}}(f) \geq 3$. With the information from Example 6, we obtain that

$$L(G - F) = \{y, y^2, xy, xy^2, x^2y\}$$

As before, the set of all rational differential forms on some curve χ is denoted by $\Omega(\chi)$. For a divisor D we define

$$\Omega(D) = \{w \in \Omega(\chi) \mid (w) - D \geq 0\}$$

The dimension of $\Omega(D)$ is denoted $\delta(D)$ and called the index of speciality. The dimension of $\Omega(D)$ is always $\delta(D) = l(W - D)$ where W is a representative of the equivalence class of canonical divisors. From the definitions above we can derive that $\Omega(X)$ is always non-zero for a canonical divisor.

Another important class of divisors in the search for decoding algorithms for algebraic geometry codes is what we call special divisors. Let χ be an algebraic curve and W its equivalence class of canonical divisors, then a special divisor is defined as follows

Definition 9. A divisor E on χ is called *special* if it is effective and $l(W - E)$ is non zero.

It follows from this definition that a special divisor is linearly equivalent to $W - D$ as long as D is any other effective divisor of χ. The next theorem gives an upper bound for a special divisor

Theorem 4 (Clifford's Theorem)**.** *For a special divisor E on the curve χ the following hold*

$$\frac{\deg(E)}{2} - (l(E) - 1) \geq 0 \tag{3.2}$$

To find a lower bound on the dimension of a special divisor on a curve we define Clifford's defect

Definition 10. For a curve χ, let ε define a finite set of divisors on this curve. The *Clifford's defect* $\sigma(\varepsilon)$ of the set ε is defined as

$$\sigma(\varepsilon) = \max_{E \in \varepsilon} \{ \frac{\deg(E)}{2} - (l(E) - 1) \} \tag{3.3}$$

In this text we will consider sets $\varepsilon = \{E_0, E_1, \ldots, E_{2g-2}\}$ where E_i is a special divisor with $\deg(E_i) = i$ for $i = 0, 1, \ldots, 2g-2$. In the following way we define two subsets of ε

$$E \in \varepsilon_0 \iff \deg(E) \equiv 0 (\text{mod } 2)$$
$$E \in \varepsilon_1 \iff \deg(E) \equiv 1 (\text{mod } 2)$$

Hence we can write $\sigma_0 = \sigma(\varepsilon_0)$ and $\sigma_1 = \sigma(\varepsilon_1)$.

4 Algebraic Geometry Codes

Coding theory is primarily concerned with dealing with errors introduced by noise. Here we will discuss some basic notions in coding theory and define algebraic geometry codes.

A block code C is defined to be a linear subspace of $\mathbb{F}_q^n$ for a field $\mathbb{F}$. An element $x \in C$ is called a code word and the size of the block code is the number of code words. A code word of length n over the alphabet $\mathbb{F}_q^n$ is a vector or n-tuple $(x_1, x_2, \ldots, x_n)$ where $x_i \in \mathbb{F}$ for $1 \leq i \leq n$. If C is a code of length n and dimension k as a $\mathbb{F}_q$-vector space we say that C is an $[n, k]$-code.

The minimum distance $d(C)$ of a code is an important parameter in coding theory. The minimum distance is defined as the smallest distance between two distinct code words. By finding $d(C)$ we obtain a measure of how good the code is at detecting and correcting errors. A code with $d(C) = d$ can find $d - 1$ errors and correct up to

$$t = \lfloor (d-1)/2 \rfloor \tag{4.1}$$

errors. Thus the error correcting capability of a code is determined by its minimal distance and we say that C is t-error correcting when t is defined as above, i. e. t is the error-correcting rate of the code. To create good codes this rate should be as large as possible. Hence we search for codes with minimum distance as large as possible. In other words, we search for codes where the code words differ as much as possible.

To calculate $d(C)$ we use the Hamming distance on $\mathbb{F}_q^n$. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be elements in $\mathbb{F}_q^n$. The Hamming distance between x and y is defined as $d(x, y) = \mid \{i \mid x_i \neq y_i\} \mid$. Hence the distance between two words in $\mathbb{F}_q^n$ is defined as the number of positions in which the elements differ. The minimum distance is the smallest Hamming distance between any two distinct elements of the code. Thus,

$$d(C) = \min\{ \mid \{i \mid x_i \neq y_i\} \mid \}$$

The space of which all code words of some given length are situated is the so called Hamming space. The dimension of this space is the number of digits in the code words contained in the space. The separation of the points is measured by the Hamming distance. The Hamming sphere is the set of all words in the Hamming space whose Hamming distance from some given word (then center of the sphere) does not exceed some given value (the Hamming radius).

Another notion commonly used in coding theory is the weight of an element in $\mathbb{F}_q^n$. This size is defined as the distance between a code word and the zero-element, i. e. the number of non-zero coordinates. Let $x \in \mathbb{F}_q^n$, then the weight of x is $\text{wt}(x) = d(x, 0) =\mid \{i \mid x_i \neq 0\} \mid$. From this definition we can also define the minimum distance of C to be the smallest non-zero weight.

$$d(C) = \text{wt}(C) = \min\{d(x, y) \mid x \neq y, x, y \in C\}$$

Since C is a vector space, we can assign a basis of size k to the code. If $C \subset \mathbb{F}_q^n$ is an $[n, k, d]$-code, the generator matrix G of C is a $k \times n$-matrix whose rows are elements in the basis of C. The generator matrix is a matrix whose rows are independent and span the code. For each generator matrix there exists an $(n - k) \times n$ matrix H, called a parity check matrix. This matrix is defined by

$$C = \{\mathbf{x} \in \mathbb{F}_q^n \mid H\mathbf{x}^T = 0\}$$

Clearly the parity check matrix checks whether an n-tuple in $\mathbb{F}_q^n$ is a code word of C or not. The rows of H is independent which means that the parity check matrix must be a generator matrix for another code. This code is called the dual of C and defined as

$$C_\Omega = \{u \in \mathbb{F}_q^n \mid \langle u, c\rangle = 0 \ \forall \ c \in C\}$$

where $\langle u, c\rangle = \sum_{i=1}^n u_i c_i$ is the canonical inner product of u and c in $\mathbb{F}_q^n$. The dual of C is the space of all vectors orthogonal to C. A dual of an $[n, k]$-code with generator matrix G and parity check matrix H is an $[n, n - k]$-code with generator matrix H and parity check matrix G. If the generator matrix can be written as $G = [I_k \mid P]$, the parity check matrix is $H = [-P^T \mid I_{n-k}]$. The dimension of a code and the dimension of its dual always add up to the length n and $(C_\Omega)_\Omega = C$. If $C = C_\Omega$ then C is called a self-dual code.

A problem when constructing codes is to find codes whose dimension and minimum distance are large compared with the length of the code. If the dimension of a code is large, the minimum distance should be small. Consider a $[n, k, d]$-code C. Define

$$L = \{(x_1, \ldots, x_n) \in \mathbb{F}_q^n \mid x_i = 0 \text{ for all } i \geq d\}$$

This is a linear subspace of $\mathbb{F}_q^n$. This means that every element in L has weight less than or equal to $d - 1$ and it follows that $L \cap C = 0$. Since $\dim(C) = k$ and $\dim(L) = d - 1$ we

get that

$$\begin{aligned} k + (d-1) &= \dim(C) + \dim(L) = \dim(C+L) + \dim(C \cap L) \\ &= \dim(C+L) \leq n \end{aligned}$$

which implies

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

This is what we call the singleton bound for the minimal distance and it provides an upper bound for the dimension of the code and the minimum distance. With this bound we have confirmed the relationship between the minimum distance and dimension of a code as discussed above. In general this is a weak bound. But codes satisfying $k+d = n+1$ are optimal codes and called maximum distance separable (MDS) codes. Usually it is much harder to find a lower bound for d and k than obtaining the upper bound. This leads to a discussion of algebraic geometry codes or what we also call Goppa codes. What makes Goppa codes so interesting is the fact that there exists a good lower bound for the minimum distance (and the dimension of the code).

We denote an algebraic geometry code by $C_L(D, G)$ with respect to an algebraic curve χ with divisors D and G. $C_L(D, G)$ is a set of n-tuples from the projective space of the form $(f(P_1), f(P_2), \ldots, f(P_n))$. Here $\{P_1, \ldots, P_n\}$ is a set of n fixed points, where $P_i \in \mathbb{F}_q^n$ for $i = 1, \ldots, n$ and f are functions with given poles and zeros. This is shown in the following definition:

Definition 11. Let G and $D = P_1 + \ldots + P_n$ be divisors of a curve χ where P_i are n distinct places of degree 1 such that $\text{supp}(G) \cap D = \emptyset$. Let $L(D)$ be the k-dimensional vector space which contains elements of the set $\{f \in \mathbb{F}_q^k \mid \text{div}(f) \leq D\}$.Then the *algebraic geometry code* $C_L(D, G) \in \mathbb{F}_q^n$ is defined as follows:

$$C_L(D, G) = \{(x(P_1), \ldots, x(P_n)) \mid x \in L(G)\} \tag{4.2}$$

Theorem 5. *Let G and $D = P_1 + \ldots + P_n$ be divisors on a nonsingular curve χ over $\mathbb{F}_q^k$ where the P_i's are distinct $\mathbb{F}_q$-rational points such that $D \cap supp(G) = \emptyset$. Assume that $\deg(G) < n$ so that the evaluation map $ev_{\mathcal{P}}$ from $L(G)$ to $C_L(D, G)$ is injective. Then*

1. *$C_L(D, G)$ is an $[n, k, d]$ code with $d \geq n - \deg(G)$ and $k = l(G) \geq \deg(G) + 1 - g$ which implies $k + d \geq n + 1 - g$.*

2. *If we also assume that $2g - g < \deg(G) < n$, then $k = \deg(G) + 1 - g$.*

3. If $\{f_1, \ldots, f_k\}$ is a basis for $L(G)$ then

$$A = \begin{bmatrix} f_1(P_1) & f_1(P_2) & \cdots & f_1(P_n) \\ f_2(P_1) & f_2(P_2) & \cdots & f_2(P_n) \\ \vdots & \vdots & & \vdots \\ f_k(P_1) & f_k(P_n) & \cdots & f_k(P_n) \end{bmatrix}$$

is a generator matrix for the algebraic geometry code $C_L(D, G)$

Proof. (Theorem 5) To prove the first part of the theorem note that ev_D is an surjective linear map from $L(G)$ to $C_L(D, G)$. Hence

$$\mathrm{Ker}(ev_D) = \{f \in L(G) \mid \mathrm{ord}_{P_i}(f) > 0 \ , \ i = 1, 2, \ldots, n\} = L(G - D)$$

Thus it follows that

$$k = \dim(C_L(D, G)) - \dim(L(G - D)) = l(G) - l(G - D)$$

Assume that $C_L(D, G)$ is non-zero. Choose any $f \in L(G)$ with $\mathrm{wt}(ev_D(f)) = d$. Then, exactly $n - d$ places, say $P_{i_1}, \ldots, P_{i_{n-d}}$ in $\mathrm{supp}(G)$ are zeros of f and $0 \neq f \in L(G - (P_{i_1}, \ldots, P_{i_{n-d}}))$. Since this Riemann-Roch space is non-empty the degree of the divisor must be positive:

$$\begin{aligned} 0 \leq \deg(G - (P_{i_1}, \ldots, P_{i_{n-d}})) &= \deg(G) - (\deg(P_{i_1}) + \ldots + \deg(P_{i_{n-d}})) \\ &= \deg(G) - n + d \end{aligned}$$

Which leads to $d \geq n - \deg(G)$.

Now, assume that $2g - 2 < \deg(G) < n$. For a canonical divisor W of χ, $\deg(W) = 2g - 2$. Since $\deg(G) > 2g - 2$

$$\deg(W - G) = \deg(W) - \deg(G) = 2g - 2 - \deg(G) < 0$$

By the definition of the Riemann-Roch space $L(W - G) = 0$ and $l(W - G) = 0$. Hence the Riemann-Roch theorem implies

$$l(G) = \deg(G) + 1 - g + l(W - G) = \deg(G) + 1 - g$$

We need the evaluation map $ev_{\mathcal{P}}$ to have a trivial kernel. Choose any $f \in L(G)$ so that $ev_D = 0$. It follows that$f(P_i) = 0$ for $i = 1, \ldots, n$. Thus P_i is a zero of f and the coefficient in front of P_i is non-zero in $\mathrm{div}(f)$. Since P_i is not an element in the support of G

$$\mathrm{div}(f) + G - P_1 + \ldots - P_n \geq 0$$

By the definition of the Riemann-Roch space we have $f \in L(G - P_1 + \ldots - P_n)$. However, the degree of G is less than n and $\deg(D - P_1 + \ldots - P_n) = \deg(D) - \deg(P_1 + \ldots + P_n) < 0$.

It follows that $L(G - P_1 + \ldots - P_n) = \{0\}$ and f must be the zero element. Hence the kernel of the evaluation map ev_D is zero.

Since the only the zero element is mapped to zero, the dimension of $L(G)$ is equal to the dimension of $C_L(D, G)$ and $k = \deg(G) + 1 - g$ which proves the second part of the theorem.

By definition a generator matrix of an $[n, k]$-code C is an $k \times n$-matrix whose rows form a basis for C. Since the kernel of the map $ev_{\mathcal{P}}$ is zero we get

$$ev_D(L(G)) = C_L(D, G) \simeq L(G)/\mathrm{Ker}(ev_D) = L(G)$$

An isomorphism between two vector spaces maps the basis in one vector space to the basis of the other vector space. Since $\{f_1, f_2, \ldots, f_k\}$ is a basis for $L(G)$,

$$\begin{aligned} &\{ev_D(f_1), ev_D(f_2), \ldots, ev_D(f_k)\} \\ =&\{(f_1(P_1), f_1(P_2), \ldots, f_1(P_n)), \ldots, (f_k(P_1), f_k(P_2), \ldots, f_k(P_n))\} \end{aligned}$$

is a basis for $C_L(D, G)$. And the matrix A is a generator matrix for $C_L(D, G)$ and we have proved the last part. □

Remark. We call $d^* = n - \deg(G)$ the designated minimum distance of $C_L(D, G)$. Hence we say that the designed error correction bound is $\lfloor (d^* - 1)/2 \rfloor$.

A nice example of algebraic geometry codes are the Reed-Solomon codes. These error-correcting codes were invented by Irving S. Reed and Gustave Solomon. These codes are widely used in coding theory mainly due to their "good" parameters (good lower minimal distance) and easy decoding process. What is called a Cross Interleaved Reed-Solomon code is used in CD's to make the sound perfect even though the surface may be partly destroyed or affected. The code protect against scratches, cracks and dirt. This code is quite effective and may correct up to four thousand errors which corresponds to a crack of size about two and a half millimeter. The next example shows how to build such codes.

Example 10. Let $n = q - 1$ and $t \in \mathbb{F}_q$ be a primitive element in the multiplicative group $\mathbb{F}_q^\times$. This means that $\mathbb{F}_q^\times = \{t, t^2, t^3, \ldots, t^n = 1\}$. Let $k \in \mathbb{Z}$ be such that $1 \leq k \leq n$. We then define the k-dimensional vector space L_k by $\{f \in \mathbb{F}_q[X] \mid \deg f \leq k - 1\}$. Define the evaluation map ev by

$$\begin{aligned} ev : L_k &\longrightarrow \mathbb{F}_q^n \\ ev(f) &\mapsto (f(t), f(t^2), \ldots, f(t^n)) \end{aligned}$$

This map is both $\mathbb{F}_q$ linear and injective. Hence

$$C_k = \{f(t), f(t^2), \ldots, f(t^n) \mid f \in L_k\} \tag{4.3}$$

is an $[n, k]$-code over $\mathbb{F}_q$ and is what we call a Reed-Solomon code. The weight of a code word in the Reed-Solomon code is

$$wt(c) = n- \mid \{i \in \{1, \ldots, n\} \mid f(t^i) = 0\} \mid \leq n - \deg(f) \leq n - (k-1)$$

Which implies that

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

By the singleton bound another restriction on d is $d \geq n + 1 - k$. Hence $d = n + 1 - k$ for Reed-Solomon codes which implies that all Reed-Solomon codes are MDS codes.

We denote a Reed-Solomon code by $RS(n, k)$. Each such code can correct up to t errors where t depends on n and k in the following way:

$$t = \frac{1}{2}(n - k)$$

These linear block codes are based on finite field arithmetic where each code word is generated using a generator polynomial, $g(x)$. This means that all valid code words $c(x)$ can be divided by the generator polynomial, i. e. $c(x) = j(x)g(x)$ where $j(x)$ is a polynomial called the information block. The encoder in these codes takes a block of digital data and adds some redundancy by adding extra bits to the information word. In other words the encoder takes k data symbols and adds some parity symbols to make a code word of n symbols.

Elliptic curves are well classified in terms of their isomorphism classes and structures. This make them useful when constructing algebraic geometry codes. Say $E(\mathbb{F}_q)$ is an elliptic curve over a finite field $\mathbb{F}_q$. Let $P_1, P_2, \ldots, P_n$ be $\mathbb{F}_q$-rational points and P_∞ the point at infinity. For some positive integer m we choose a divisor $G = mP_\infty$. Another divisor chosen is $D = P_1 + \ldots + P_n$. From these divisors we can build the code $C_L(D, G)$ over $\mathbb{F}_q$. This code have parameters $[n, m, d \geq n - 1]$ where $\mid n - (q + 1) \mid \leq \lfloor 2\sqrt{q} \rfloor$. In the next example we build an algebraic geometric code over an elliptic curve over the field $\mathbb{F}_{13}$

Example 11. Again we consider the the elliptic curve give by $\mathcal{E} : \{Y^2 = X^3+9X+4\} \cup \{(0 : 1 : 0)\}$ over the finite field with 13 elements. As we saw in Example 9 this curve has 13 rational points in $\mathbb{F}_{13}$ and a point a infinity P_∞. To build a code over this curve, choose the divisor $G = 8P_\infty$ and $D = P_1 + P_2 + \ldots P_{12}$. As seen before the basis of $L(G)$ is $\{1, x, x^2, x^4, y, y^2, xy, x^2y\}$. Thus the code $C_L(D, G)$ is constructed by the evaluation map

$$\begin{aligned} ev_D : L(G) &\longrightarrow \mathbb{F}_q^n \\ x \in L(G) &\mapsto (x(P_1), x(P_2), \ldots, x(P_{12})) \end{aligned}$$

From this we can calculate generator matrix A for the code:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 2 & 2 & 6 & 8 & 6 & 8 & 11 & 11 \\ 0 & 0 & 1 & 1 & 4 & 4 & 10 & 12 & 10 & 12 & 4 & 4 \\ 0 & 0 & 1 & 1 & 3 & 3 & 9 & 1 & 9 & 1 & 3 & 3 \\ 2 & 11 & 1 & 12 & 2 & 11 & 1 & 4 & 12 & 9 & 2 & 11 \\ 4 & 4 & 1 & 1 & 4 & 4 & 1 & 3 & 1 & 3 & 4 & 4 \\ 0 & 0 & 1 & 12 & 4 & 9 & 6 & 6 & 7 & 7 & 9 & 4 \\ 0 & 0 & 1 & 12 & 8 & 5 & 10 & 9 & 3 & 4 & 8 & 5 \end{bmatrix}$$

Here the first row of the matrix is 1 (first element of the basis of $L(G)$) evaluated in $P_1, \ldots, P_n$, the second row is x (second element of the basis of $L(G)$) evaluated in $P_1, \ldots, P_n$, the third row is x^2 (third element of the basis of $L(G)$) evaluated in $P_1, \ldots, P_n$ etc.

The length of $C_L(D, G)$ is $n = 12$, the dimension is $k = 8$ and designed minimum distance $d^* = 4$. By calculating the Gauss-Jordan form of A, A can be written on the form $[I_k \mid P]$. It follows that the parity check matrix of $C_L(D, G)$ is

$$H = [-P^T \mid I_{n-k}] = \begin{bmatrix} 5 & 8 & 2 & 11 & 1 & 12 & 12 & 0 & 1 & 0 & 0 & 0 \\ 3 & 10 & 3 & 10 & 4 & 9 & 0 & 12 & 0 & 1 & 0 & 0 \\ 2 & 1 & 11 & 11 & 1 & 2 & 1 & 9 & 0 & 0 & 1 & 0 \\ 8 & 8 & 8 & 1 & 4 & 12 & 1 & 9 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It is easy to verify that $A \cdot H^T = 0$.

5 Decoding of BCH Codes

In this we give section a short discussion of cyclic codes. An example of such codes are the BCH codes. One of the advantages with BCH codes is that they are easy to decode. This is done by a method called syndrome decoding. There exist several different algorithm for decoding BCH codes which use syndrome decoding. Here we show how to decode BCH codes by using the theory of linear feedback shift registers (LFSR).

5.1 Cyclic Codes and BCH Codes

A code C of length n over $\mathbb{F}_q$ is cyclic if for each code word $c = c_0c_1 \ldots c_{n-2}c_{n-1}$ in C, $c_{n-1}c_0c_1 \ldots c_{n-2}$ is also a code word of the same code. This means that all cyclic shifts of a code word are elements in the same cyclic code. The vector $c_{n-1}c_0c_1 \ldots c_{n-2}$ is obtained from c by shifting the coordinates $i \mapsto i + 1 (\text{mod } n)$. Usually we identify the vector space $\mathbb{F}_q^n$ with the polynomial vector space $\mathbb{F}_q^n[x]$ since there is an isomorphic correspondence between vectors in $\mathbb{F}_q$ and polynomials in $\mathbb{F}_q[x]$. Thus

$$c = c_0c_1 \ldots c_{n-2}c_{n-1} \longleftrightarrow c(x) = c_0 + c_1x + \ldots + c_{n-2}c^{n-2} + c_{n-1}c^{n-1}$$

The cyclic shifts in $\mathbb{F}_q[x]$ are defined with multiplication by modulo $(x^n - 1)$. By

$$\begin{aligned} xc(x) &= c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1} + c_{n-1}x^n \mathrm{mod}(x^n - 1) \\ &= c_{n-1} + c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1} \end{aligned}$$

$xc(x)$ is also in C. This leads to the study of codes in the residue class $\mathbb{F}_q[x]/(x^n-1)$. Since the finite field $\mathbb{F}_q$ is a PID (principal ideal domain), ideals in $\mathbb{F}_q[x]/(x^n - 1)$ are principal. Thus cyclic codes can be seen as principal ideals in $\mathbb{F}_q[x]/(x^n - 1)$.

BCH codes or Bose-Ray-Chaudhuri-Hocquenghem codes are cyclic codes which still (to a certain degree) are used in practice. The motivation for the discovery of BCH was the need to construct a cyclic code C over $\mathbb{F}_q$ of length n with similarly high minimum distance and high dimension. These codes have a reliable lower bound on the minimum distance called the BCH bound (which also provide us with a lower bound for other cyclic codes).

Proposition 4. *Let C be a cyclic code over $\mathbb{F}_q$ of length n and suppose that the minimum distance of the code is d. Let T be the defining set of C containing $\delta - 1$ elements for $\delta \in \mathbb{Z}$. Then $d \geq \delta$*

For proof and existence of the BCH bound see Section 5 of Chapter 4 in [4].

A BCH code with designed distance δ is defined to be the cyclic code of length n over $\mathbb{F}_q$ where the generator polynomial $g(x)$ is the least common multiple of the minimal polynomials $\alpha^l, \alpha^{l+1}, \ldots, \alpha^{l+\delta-2}$ for some $l \in \mathbb{Z}$ where α is a n^{th} root of unity. A cyclic code with l equal to 1 is called a narrow-sense BCH code . The minimum distance of a BCH code with designed distance δ is at least δ.

5.2 The Berlekamp-Massey Decoding Algorithm

BCH codes are simple cyclic codes over a finite field $\mathbb{F}_q$, easily constructed by help the of a generator polynomial as shown in Section 5.1. Several algorithms for decoding these codes exist. For example the Euclidean algorithm discovered by Sugiyama. For an overview of this algorithm see [23]. We derive the Berlekamp-Massey algorithm for decoding BCH codes using the theory of linear feedback shift registers.

The standard Berlekamp-Massey algorithm computes a minimal polynomial $P(x)$ of a linearly recurrent sequence, $S(x) = \sum_{i=0}^{\infty} a_i x^i$ where a_i is an element of an arbitrary field. If the minimal polynomial is $P(x) = \sum_{i=0}^{d} p_i x^i$ then $P(x)$ is the smallest polynomial such that $\sum_{i=0}^{d} p_j x^{j+i}$ is zero for all $j \in \mathbb{N}$. To find the smallest such polynomial we use some of the theory behind LFSR's.

5.2.1 Linear Feedback Shift Register

Before deriving the Berlekamp-Massey decoding algorithm we look into linear feedback shift register (LFSR). The problem of finding the shortest LFSR that generates a given finite sequence is studied.

An LSFR consists of several parts or steps. First we have the shift register of length, say l. This consists of a row of l registers or what we call memory cells. We label them from left to right by $R_l, R_{l-1}, \ldots, R_2, R_1$. Each cell is capable of storing one bit. Let σ_i be the value in the i^{th} cell (from the right). To control movement between the memory cells, an electronic clock is used. The first pulse of the clock moves the value σ_l in the left-most entry R_l to R_{l-1}. The value from this register is then moved to R_{l-2} and so on. It stops when σ_2 is moved to the register currently holding σ_1 and σ_1 gets tapped to the output sequence. The cell R_l is thus left empty. For a value to fill this register we require that this value is a linear combination of the values σ_j for $j = 1, 2, \ldots, l$.

For this to work, we need what is called a tap sequence. This is an l-tuple of bits which will be denoted $(a_l, a_{l-1}, \ldots, a_2, a_1)$. We let $a_1 = 1$ and form

$$\sigma_{l+1} = \sum_{j=1}^{l} a_j \sigma_j$$

This is what we call the linear feedback. When calculated, σ_{l+1} is placed in the memory cell R_j. The initial state $\sigma_l, \sigma_{l-1}, \ldots, \sigma_2, \sigma_1$ is transformed to $\sigma_{l+1}, \sigma_l, \ldots, \sigma_3, \sigma_2$. In this step our input is σ_{l+1} and output is σ_1

An LFSR's m^{th} state is denoted by s_m. This is the bit string containing the values of all registers R_j for $j = 1, 2, \ldots, l$ after m clock pulses. The initial state, or the seed, (which is always different from the zero vector) is given by the following l-tuple:

$$s_0 = (\sigma_l \sigma_{l-1} \ldots \sigma_2 \sigma_1)$$

The state after one clock pulse is $s_1 = (\sigma_{l+1} \sigma_l \ldots \sigma_3 \sigma_2)$. In general a state $m \in \mathbb{N}$ is given by

$$s_m = (\sigma_{m+l} \sigma_{m+l-1} \ldots \sigma_{m+1} \sigma_m)$$

with linear feedback

$$\sigma_{m+l+1} = \sum_{j=1}^{l} a_j \sigma_{m+j}$$

The above is what we call a binary recurrence relation of length l.

The method above can be translated into matrix operations. Let C be what we call a "tap matrix" and S_m denote the "state matrix" defined in the following way:

$$C = \begin{bmatrix} a_1 & a_2 & a_3 & \ldots & a_{l-1} & a_l \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \quad \text{and} \quad S_m = \begin{bmatrix} \sigma_{m+l-1} \\ \sigma_{m+l-2} \\ \vdots \\ \sigma_{m+1} \\ \sigma_m \end{bmatrix} \tag{5.1}$$

Thus we can calculate the state s_{m+1} from s_m by the equality $CS_m = S_{m+1}$.

After introducing the reader to the steps of an LFSR, we now show how to find the shortest LFSR which generates a given sequence. Let $a_1, \ldots, a_N$ be a sequence of length N. We are seeking a LFSR of length L such that the first N outputs of the LFSR starting from the initial state $a_L, \ldots, a_1$ are exactly the sequence given above and L is as small as possible. Let $(L, \sigma(x))$ denote the LFSR of length L with connection polynomial

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \ldots + \sigma_L x^L \tag{5.2}$$

Let $a_1, \ldots, a_n$ with $n \leq N$ be the first n entries of the sequence of length N. Then the shortest LFSR which generates $a_1, \ldots, a_n$ is $(L(n), \sigma^{(n)}(x))$. In this section, a technique for finding the LFRS $(L(n+1), \sigma^{(n+1)}(x))$ from the LFRS $(L(i), \sigma^{(i)}(x)$ for $i \leq n$ is developed. For further use let us write $\sigma^{(n)}$ as

$$\sigma^{(n)}(x) = 1 + \sigma_1^{(n)} x + \ldots + \sigma_l^{(n)} x^l \text{ where } l = L(n)$$

The parameters $l = L(n)$ and $\sigma_1^{(n)}$, ..., $\sigma_l^{(n)}$ are unknown. These are determined by solving the following set of equations such that l is as small as possible.

$$\begin{bmatrix} a_1 & a_2 & \ldots & a_{l+1} \\ a_2 & a_3 & \ldots & a_{l+2} \\ \vdots & \vdots & & \vdots \\ a_{n-l} & a_{n-l+1} & \ldots & a_n \end{bmatrix} \cdot \begin{bmatrix} \sigma_l^{(n)} \\ \vdots \\ \sigma_1^{(n)} \\ 1 \end{bmatrix} = 0 \tag{5.3}$$

We denote the coefficient matrix above as $A(n-l, l+1)$. This is an $(n-l) \times (l+1)$ Hankel matrix. To choose such an l may not always be straight forward and we need to work out the steps on how to proceed.

First we prove the following result.

Lemma 1. *$A(n/2, n/2)$ is nonsingular if and only if $L(n) = n/2$*

Proof. Let $n = 2l$. We will prove that $A(l, l)$ is nonsingular if $L(2l) = l$ by a contradiction. Assume that the $(m+1)^{\text{th}}$ row of $A(l, l)$ is a linear combination of the first m rows $(m < l)$, i. e. $A(l, l)$ is singular. By using that $n = 2l$ in (5.3) we get

$$\begin{bmatrix} a_1 & a_2 & \ldots & a_{l+1} \\ a_2 & a_3 & \ldots & a_{l+2} \\ \vdots & \vdots & & \vdots \\ a_l & a_{l+1} & \ldots & a_{2l} \end{bmatrix} \cdot \begin{bmatrix} \sigma_l^{(2l)} \\ \vdots \\ \sigma_1^{(2l)} \\ 1 \end{bmatrix} = 0$$

Since the $(m+1)^{\text{th}}$ row is a linear combination of the previous rows, $L(2l) \leq m < l$. This contradicts the fact that $L(2l) = l$ and hence $A(l, l)$ must be non-singular.

For the other direction, assume that $A(l,l)$ is non-singular. Hence all rows of $A(l,l)$ are linear independent of each other. If $n = 2l$ and $A(l,l)$ is non-singular we can solve (5.3) uniquely for $\sigma_1^{(n)}, \ldots, \sigma_l^{(n)}$. Hence, it is obvious that $L(2l) \leq l$. This must be an equality since $L(2l) = m < l$ implies that the $(m+1)^{\text{th}}$ row can be written as a linear combination of the m first rows of the $l \times l$-matrix. Thus we have proved the lemma. □

The next result is derived from Lemma 1 and states the uniqueness of an LFSR.

Corollary 1. *If $L(n)$ is less than or equal to $n/2$ then the LFSR $(L(n), \sigma^{(n)}(x))$ is unique, i. e. if $L(n) = m < n/2$ then $(L(n), \sigma^{(n)}(x)) = (m, \sigma^{(2m)}(x))$*

Note that for $L(n) > n/2$ the LFSR $(L(n), \sigma^{(n)(x)})$ is no longer unique. It can be shown that it has a freedom $2L(n) - 1$.

The function $L(n)$ is a non-decreasing stepwise function of n. Thus it may be helpful to know at which points this function jumps. Assume that $A(l,l)$ is a nonsingular matrix and increase n from $2l$. Assume that $L(n) = l$ for $2l \leq n \leq 2l + k - 1$ and $L(2l+k) > l$ for $k \geq 1$. With these assumptions in mind we derive that

$$A(l+k, l+1)[\sigma_l, \ldots, \sigma_1, 1]^T = [0, \ldots, 0, d]^T \text{ for } d \neq 0$$

We want to show that $L(2k+2l) = k + l$. From Lemma 1 this holds when the matrix $A(l+k, l+k)$ is non-singular. We have that

$$A(l+k,l+k)\left[\begin{array}{cccc|cccc} 1 & 0 & \ldots & 0 & \sigma_l & 0 & \ldots & 0 \\ 0 & \ddots & & 0 & \vdots & \sigma_l & & \vdots \\ 0 & & \ddots & 0 & \vdots & \vdots & \ddots & 0 \\ 0 & \ldots & 0 & 1 & \sigma_1 & \vdots & \ddots & \sigma_l \\ \hline 0 & \ldots & \ldots & 0 & 1 & \sigma_1 & \ddots & \vdots \\ \vdots & \ddots & & 0 & 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & 0 & 0 & 0 & \ddots & \sigma_1 \\ 0 & \ldots & \ldots & 0 & 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{c|ccc} & 0 & \ldots & 0 \\ A(l,l) & \vdots & \ddots & \vdots \\ & 0 & \ldots & 0 \\ \hline & 0 & \ldots & d \\ & \vdots & \iddots & \\ & d & & \end{array}\right]$$

which implies that $A(l+k, l+k)$ is non-singular and by Lemma 1 $L(2l+2k)$ must be $l+k$.

Next we prove that $L(2l+k) = l+k$. Since $L(2l+2k) = l+k$, $L(2l+k)$ must be less than or equal to $l+k$ (this follows from the fact that L is an increasing function). Assume that $l < L(2l+k) = m < l+k$. We then have that

$$A(2l+k-m, m+1)[\hat{\sigma}_m, \ldots, \hat{\sigma}_1, 1]^T = 0 \tag{5.4}$$

By multiplying (5.4) from the left by $[0, \ldots, 0, \sigma_l, \ldots, \sigma_1, 1]$ we get that

$$d = [0, \ldots, 0, d][\hat{\sigma}_m, \hat{\sigma}_1, 1]^T = 0$$

which contradicts the fact that d is non-zero. Thus $L(2l + k) = l + k$ and

$$L(n) = l + k \text{ for } 2l + k \leq n \leq 2l + 2k$$

By the above discussion we obtain then next lemma.

Lemma 2. *Let $n = 2l'$ and $n = 2l$ under the condition that $l' < l$ be the two points where the function $L(n)$ becomes $n/2$. Thus, the point $l + l'$ is the only point in the interval $2l' \leq n \leq 2l$ where $L(n)$ jumps. At this point the length jumps with $l - l'$ which is the same as half of the length of the interval.*

Remark. The stepwise function $L(n)$ is not to be confused with $L(D)$, the Riemann-Roch space of a divisor D.

5.2.2 Berlekamp-Massey

The algorithm derived in this section is an effective method of finding the shortest linear feedback shift register of a given sequence. We find an algorithm that calculates the LFSR $(L(n+1), \sigma^{(n+1)}(x))$ from LFSR's $(L(i), \sigma^{(i)}(x))$ when $\sigma^{(n+1)}(x) \neq \sigma^{(n)}(x)$.

From Corollary 1 the above can only happen when $L(n+1) > (n+1)/2$. Thus, consider an n in $\mathbb{N}$ such that $l + l' - 1 \leq n < 2l$. Let $d(n)$ be the difference between a_{n+1} and the $(n+1)$th output of the LFSR $(L(n), \sigma^{(n)}(x))$. Denote the most recent point where there was an length change prior to n by $k(n)$.

Algorithm 1 (Standard Berlekamp-Massey Algorithm)**.** Initial conditions: If $a_1 \neq 0$ then $L(0) = 0$ and $L(1) = 1$. In this case $k(1) = 0$, $d(0) = a_1$, $\sigma^{(0)}(x) = 1$ and $\sigma^{(1)}(x) = 1$. On the contrary, if $a_1 = \ldots = a_{m-1} = 0$ and $a_m \neq 0$ then $L(0) = L(1) = \ldots = L(m-1) = 0$ while $L(m) = m$. Other conditions in this case are $k(m) = m - 1$, $d(m-1) = a_m$, $\sigma^{(0)}(x) = \ldots = \sigma^{(m-1)}(x) = 1$ and $\sigma^{(m)}(x) = 1$.

1. Compute $d(n)$ by $d(n) = a_{n+1} + \sigma_1^{(n)} a_n + \ldots + \sigma_{L(n)}^{(n)} a_{n+1-L(n)}$.
2. If $d(n)$ is 0 let $L(n) = L(n+1)$, $k(n+1) = k(n)$ and $\sigma^{(n+1)}(x) = \sigma^{(n)}(x)$.
3. If $d(n) \neq 0$ calculate $\Delta L = n + 1 - 2L(n)$.
4. If $\Delta L > 0$ then $L(n+1) = L(n) + \Delta L$ and $k(n+1) = n$.
5. If $\Delta L \leq 0$ then $L(n+1) = L(n)$ and $k(n+1) = k(n)$.

6. Update the connection polynomial by

$$\sigma^{(n+1)}(x) = \sigma^{(n)}(x) - (d(n)/d(k(n)))x^{(n-k(n))}\sigma^{(k(n))}(x)$$

To show the existence of the Berlekamp-Massey algorithm let $n = l + l' - 1$. Since $L(n) = l' < l = L(n+1)$ it is possible that $\sigma^{(l+l')}(x) \neq \sigma^{(l+l'+1)}(x)$. Let

$$\sigma^{(l+l')}(x) = 1 + \sigma_1 x + \ldots + \sigma_l x^l \text{ and } \sigma^{(l+l'-1)}(x) = 1 + \hat{\sigma}_1 x + \ldots + \hat{\sigma}_l x^l$$

The above equations are solved by calculating

$$A(l', l+1)[\sigma_l, \ldots, \sigma_1, 1]^T = 0 \tag{5.5}$$

and

$$A(l, l')[\hat{\sigma}_{l'}, \ldots, \hat{\sigma}_1, 1]^T = [0, \ldots, 0, d'] \text{ for } d' \neq 0 \tag{5.6}$$

By considering only the last l' equations this system of equations can be rewritten as:

$$A(l', l+1)[0, \ldots, 0, \hat{\sigma}_{l'}, \ldots, \hat{\sigma}_1, 1]^T = [0, \ldots, 0, d]^T \tag{5.7}$$

Let $l" \leq l'$ be so that in the interval $[2l", 2l"]$, $L(n)$ jumps at $l'' + l'$. The points in the interval where $L(n)$ becomes $n/2$ is $2l"$ and $2l'$. By comparing (5.5) and (5.7) we derive the following equation

$$\sigma^{(l"+l'-1)}(x) = 1 + \tilde{\sigma}_1 x + \ldots + \tilde{\sigma}_{l"} x^{l"}$$

which implies

$$A(l', l"+1)[\tilde{\sigma}_{l"}, \ldots, \tilde{\sigma}_1, 1]^T = [0, \ldots, 0, d"]^T$$

These set of equations can be rewritten as

$$A(l', l+1)[\tilde{\sigma}_{l"}, \ldots, \tilde{\sigma}_1, 1, 0, \ldots, 0]^T = [0, \ldots, 0, d"]^T \text{ for } d" \neq 0$$

From the equations above and (5.7) we obtain the following vector, which satisfies (5.5):

$$[\sigma_l, \ldots, \sigma_1, 1]^T = [0, \ldots, 0, \hat{\sigma}_{l'}, \ldots, \hat{\sigma}_1, 1]^T - (d'/d")[\tilde{\sigma}_{l"}, \ldots, \tilde{\sigma}_1, 0, \ldots, 0]^T$$

By transforming this vector into a connection polynomial we get

$$\sigma^{(l+l')}(x) = \sigma^{(l+l'-1)}(x) - (d'/d")x^{l-l"}\sigma^{(l"+l'-1)}(x) \tag{5.8}$$

Let us consider the assumption that $n = l + l' + k - 1$ (or $n + 1 = l + l' + k$) where k is in the interval $[1, l - l']$. We know that $L(n) = L(n+1)$. But there is no guarantee that $\sigma^{(n)} = \sigma^{(n+1)}$. To find the relation ship between these two connection polynomials write

$$\sigma^{(l+l'+k)}(x) = 1 + \sigma_1 x + \ldots + \sigma_l x^l \tag{5.9}$$

$$\sigma^{(l+l'+k-1)}(x) = 1 + \sigma_1^0 x + \ldots + \sigma_l^0 x^l \tag{5.10}$$

The connection polynomial in (5.9) must satisfy $A(l' + k, l + 1)[\sigma_l, \ldots, \sigma_1, 1]^T = 0$. In a similar way (5.10) must satisfy $A(l' + k, l + 1)[\sigma_l^0, \ldots, \sigma_1^0, 1]^T = [0, \ldots, 0, d]^T$. In this case d may or may not be equal to zero. If d is zero then the two connection polynomials are equal. If $d \neq 0$ then $\sigma^{(l+l'+k)}(x) \neq \sigma^{(l+l'+k-1)}(x)$ and we need a formula to determine $\sigma^{(l+l'+k)}(x)$.

Since $l > l' + k$, we write the lower $l' + k$ equations of (5.6)

$$A(l' + k, l + 1)[0, \ldots, 0, \hat{\sigma}_l, \ldots, \hat{\sigma}_1, 1, 0, \ldots, 0]^T = [0, \ldots, 0, d]^T$$

Thus we can find $\sigma^{(l+l'+k)}$ by

$$\sigma^{(l+l'+k)} = \sigma^{(l+l'+k-1)}(x) - (d/d')x^k \sigma^{(l+l'-1)} \tag{5.11}$$

From the discussion in this section we have two formulas for updating the LFSR, equation (5.8) and (5.11). These formulas leads to the existence of Algorithm 1.

Remark. The running time for the Berlekamp-Massey algorithm for determining the linear complexity of a length n sequence is $\mathcal{O}(n^2)$. The algorithm delivers a unique connection polynomial if and only if the length L of the $LFSR$ is less than or equal to $n/2$. This decoding algorithm can be adapted to decode algebraic geometry codes. For more information on the subject see [28].

6 The Basic Decoding Algorithm

In this section a decoding algorithm for AG codes will be constructed. This is done by choosing a suitable divisor so that an error-locator function can be found. We find that the positions of the errors which occur during transmission are among the zeros of this function. As long as the error function does not have to many zeros, the error vector can be determined, and it is possible to decode the received word.

For simplicity we assume there are no erasures (errors with know position). Consider a nonsingular curve of genus g with divisors $D = P_1 + P_2 + \ldots + P_n$ and G such that $\text{supp}(G) \cap D = \emptyset$. From these divisors we build a code $C_L(D, G)$ with the evaluation map

$$\begin{aligned} ev_D : L(G) &\longrightarrow \mathbb{F}_q^n \\ x \in L(G) &\mapsto (x(P_1), x(P_2), \ldots, x(P_n)) \end{aligned}$$

The image of ev_D is $C_L(D,G)$. This is a $\mathbb{F}_q$-linear $[n,k,d]$-code with length n (the number of places in D), minimum distance d equal to or greater than $n-\deg(G)$ and dimension $k \geq \deg(G)+1-g$. Thus the designed minimum distance is $d^* = n-\deg(G)$. Recall that the dual of $C_L(D,G)$ is itself a $\mathbb{F}_q$-linear code defined by

$$C_\Omega(D,G) = C_\Omega(D,G) = \{b \in \mathbb{F}_q^n \mid (b \cdot h) = 0 \quad \forall\ h \in L(G)\}$$

The syndrome $(b \cdot h)$ is defined as $\sum_{\nu=1}^{n} b_\nu h(P_\nu)$ where $b = (b_1, \ldots, b_n)$ is an element of $\mathbb{F}_q^n$ and h is an element of $L(G)$ (the Riemann-Roch space of G). The dual code has parameters $[n_\Omega, k_\Omega, d_\Omega]$ satisfying $n_\Omega = n$, $d_\Omega \geq \deg(G) - 2g + 2$ and $k_\Omega = n - k \geq n - \deg(G) - 1 + g$.

Assume that $a \in \mathbb{F}_q^n$ is a received word so that $a = c + e$. Here c is a code word contained in $C_\Omega(D,G)$ and e is an element of $\mathbb{F}_q^n$ called the error vector. The error vector contains t errors such that

$$\mathrm{wt}(e) = t \leq \frac{1}{2}(d-1)$$

The set of error positions is defined as

$$I = \{\nu \mid 1 \leq \nu \leq n \text{ and } e_\nu \neq 0\}$$

It follows that the cardinality of I is equal to the weight of e and hence have an upper bound of t.

To construct the decoding algorithm the first step is to choose a divisor F disjoint from G satisfying the three conditions below

$$\mathrm{supp}(F) \cap \mathrm{supp}(D) = \emptyset \tag{6.1}$$

$$\deg(F) < \deg(G) - (2g-2) - t \tag{6.2}$$

$$l(F) > t \tag{6.3}$$

The vector spaces $L(F)$, $L(G-F)$ and $L(G)$ are important for the existence of the basic decoding algorithm. Denote the bases of the vector spaces $\{f_1, \ldots, f_l\}$, $\{g_1, \ldots, g_k\}$ and $\{h_1, \ldots, h_m\}$ respectively.

From the definition of Riemann-Roch spaces we show that $f_\lambda g_\rho$ for $1 \leq \lambda \leq l$ and $1 \leq \rho \leq k$ must be elements of $L(G)$. Let $F = \sum n_P P$ and $G = \sum m_P P$, then

$$L(F) = \{f \mid \mathrm{ord}_P(f) \geq -n_p\} \text{ and } L(G-F) = \{g \mid \mathrm{ord}_P(g) \geq -(m_P - n_P)\}$$

It follows that

$$\mathrm{ord}_P(f_\lambda g_\rho) = \mathrm{ord}_P(f_\lambda) + \mathrm{ord}_P(g_\rho) \geq -n_P - (m_P - n_P) = -m_P$$

which shows that $f_\lambda g_\rho$ is an element of $L(G)$. Hence $ev_D(f_\lambda g_\rho)$ for $1 \leq \lambda \leq l$ and $1 \leq \rho \leq k$ is an element of $C_L(D, G)$. Since all elements of $C_L(D, G)$ are orthogonal to elements of its dual, $ev_D(f_\lambda g_\rho)\mathbf{x}^T = 0$ for $\mathbf{x}$ in $C_\Omega(D, G)$. Moreover, if f is an element of the basis of $L(G)$ and zero at all error positions and g is an element of $L(G - F)$ then

$$\sum a_i f(P_i)g(P_i) = \sum e_i f(P_i)g(P_i) = \sum_{i \in I} e_i f(P_i)g(P_i) = 0$$

Define the kernel of a received word a and a divisor F as

$$K(F, a) = \{f \in L(F) \mid \sum a_i f(P_i)g(P_i) = 0 \ \ \forall g \in L(G - F)\}$$

By definition the kernel contains all error locator functions of $L(F)$. Let $\sum_{\nu \in I} P_\nu$ be the divisor of error positions. Thus $L(F - \sum_{\nu \in I} P_\nu)$ is the vector space of error locator functions which implies

$$L(F - \sum_{\nu \in I} P_\nu) \subseteq K(F, a)$$

Since $\mid I \mid \leq t$ and $l(F) > t$, F is more effective than $\sum_{\nu \in I} P_\nu$, which implies that $l(F - \sum_{\nu \in I} P_\nu) \neq 0$. Hence $L(F - \sum_{\nu \in I} P_\nu) \neq 0$ as long as $l(F) \geq t + 1$ which again hold when $\deg(F) \geq t + g$.

Moreover, we use the assumption that $\deg(G - F) > t + 2g - 2$. With this lower bound on the degree of $G - F$, it can be shown that $C_\Omega(\sum_{\nu \in I} P_\nu, G - F) = 0$. For an f in the kernel of F and a received word a,

$$0 = \sum a_\nu f(P_\nu)g(P_\nu) = \sum_{\nu \in I} e_\nu f(P_\nu)g(P_\nu) \ \ \forall g \in L(G - F)$$

Let the word $w_\nu = e_\nu f(P_\nu)$ be in the dual of $C_L(\sum_{\nu \in I} P_\nu, G - F)$. Since this code is zero, $e_{\nu \in I} f(P_\nu)$ is zero for all elements in I. Hence f is zero for all error positions which implies that $L(F - \sum_{\nu \in I} P_\nu) = K(F, a)$.

To obtain that $L(F - \sum_{\nu \in I} P_\nu)$ is equal to the kernel of F and a we used the assumptions that $\deg(F) \geq t + g$ and $\deg(G - F) > t - 2g + 2$. These assumptions contradict each other when t is too large. By comparing the inequalities we obtain that the largest possible value for t must be

$$t = \left\lfloor \frac{d^* - 1 - g}{2} \right\rfloor$$

The first step of the basic decoding algorithm is to calculate what is called an error locator function. This is a non-zero function γ in $L(F)$ with the property $\gamma(P_\nu) = 0$ for all ν in I. We define the set of zeros of the function γ as

$$N(\gamma) = \{\nu \mid 1 \leq \nu \leq n \text{ and } \gamma(P_\nu) = 0\} \tag{6.4}$$

Thus the error positions are contained in $N(\gamma)$. In other words, the error positions are among the zeros of the error locator function γ. As $e_\nu = 0$ for $\nu \notin N(\gamma)$, the error vector e can be calculated when the error locator function is known.

To find the error locator function we study the following system of linear equations

$$\sum_{\lambda=1}^{l}(a \cdot f_\lambda g_\rho)x_\lambda = \sum_{\lambda=1}^{l}\sum_{\nu=1}^{n}(a_\nu f_\lambda(P_\nu)g_\rho(P_\nu))x_\lambda = 0 \quad \text{for} \quad \rho = 1, \ldots, k \tag{6.5}$$

With the knowledge of the error positions $\nu \in I$ and the solution of the above system we can create the error locator functions. This is shown in the next proposition.

Proposition. *Assume that the conditions on the divisors F, G and $G - F$ are as before and equation (6.5) has a non-trivial solution $\alpha = (\alpha_1, \ldots, \alpha_l)$. Define*

$$\gamma = \sum_{\lambda=1}^{l}\alpha_\lambda f_\lambda \in L(F) \tag{6.6}$$

Then $\gamma(P_\nu)$ is zero for all ν in I. In other words, γ is an error locator function.

Proof. Choose an element γ of $L(F - \sum_{\nu\in I} P_\nu)$ and define $\gamma = \sum_{\lambda=1}^{l}\alpha_\lambda f_\lambda$ for α_λ in $\mathbb{F}_q$. Clearly γg_ρ is an element of $L(G)$ for $1 \leq \rho \leq k$. Thus we obtain the syndom

$$(a \cdot zg_\rho) = \sum_{\lambda=1}^{l}(a \cdot f_\lambda g_\rho)\alpha_\lambda \tag{6.7}$$

Since $a = c + e$ where c is an element of the dual code of $C_L(D, G)$, $(c \cdot f) = 0$ for all $f \in L(G)$. This implies that $(c \cdot \gamma g_\rho) = 0$. Since γ is an element of $L(F - \sum_{\nu\in I} P_\nu)$, γ evaluates to zero at P_ν for $\nu \in I$, i. e. $\gamma(P_\nu) = 0$ for all $\nu \in I$. If we add the fact that $e_\nu = 0$ for $\nu \notin I$ it follows that

$$(a \cdot \gamma g_\rho) = (c + e \cdot \gamma g_\rho) = (e \cdot \gamma g_\rho) = \sum_{\nu=1}^{n} e_\nu \gamma(P_\nu)g_\rho(P_\nu) = 0 \tag{6.8}$$

Form equation (6.7) and equation (6.8), it follows that $(\alpha_1, \ldots, \alpha_l)$ is a non-trivial solution of (6.5).

Now, assume that $(z_1, \ldots, z_l)$ is an arbitrary, non-trivial solution of (6.5) and define $\gamma = \sum_{\lambda=1}^{l} z_\lambda f_\lambda$. Suppose there exists an error position $\nu_0 \in I$ such that $\gamma(P_{\nu_0}) = \sum_{\lambda=1}^{l} z_\lambda f_\lambda(P_{\nu_0}) \neq 0$. By our original assumptions

$$\deg(G - F - \sum_{\nu\in I} P_\nu) \geq \deg(G) - \deg(F) - t > 2g - 2$$

Hence $L(G - F - \sum_{\nu \in I} P_\nu) \subsetneq L(G - F - \sum_{\nu \in I \setminus \nu_0} P_\nu)$. Thus we can find an element $q \in L(G - F)$ such that $q(P_{\nu_0}) \neq 0$ and $q(P_\nu) = 0$ for all $\nu \in I \setminus \nu_0$. From this we obtain that

$$(a \cdot \gamma q) = (e \cdot \gamma q) = \sum_{\nu=1}^{n} e_\nu \gamma(P_\nu) q(P_\nu) = e_{\nu_0} \gamma(P_{\nu_0}) q(P_{\nu_0}) \neq 0 \tag{6.9}$$

Here q is a linear combination of $g_1, \ldots g_k$ and $(a \cdot \gamma g_\rho) = \sum (a \cdot f_\lambda g_\rho)\alpha_\lambda = 0$. Hence $(\alpha_1, \ldots, \alpha_l)$ is a solution of (6.5) which contradicts (6.9) and no such ν_0 exists. □

Next we determine the error values. This is done by considering the syndrome

$$\sum_{\nu \in N(\gamma)} h_\mu(P_\nu) e_\nu = (a \cdot h_\mu) \tag{6.10}$$

How the error vector is retrived from the linear system above is proven in the following proposition.

Proposition. *The error vector $(e_\nu)_{\nu \in N(\gamma)}$ is the unique solution of equation (6.10)*

Proof. Let $h_\mu \in L(G)$. We then get that

$$(a \cdot h_\mu) = (c + e \cdot h_\mu) = (e \cdot h_\mu) = \sum_{\nu=1}^{n} e_\nu h_\mu(P_\nu) = \sum_{\nu \in N(f)} h_\mu(P_\nu) e_\nu$$

As mentioned before, e_ν is equal to zero for ν not contained in the set $N(\gamma)$. Hence $(e_\nu)_{\nu \in N(\gamma)}$ is a solution of (6.10). To see that this is an unique solution, assume that $(b_\nu)_{\nu \in N(\gamma)}$ is another such solution of (6.10). Define $b = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$ such that $b_\nu = 0$ for $\nu \notin N(\gamma)$. Thus,

$$(b \cdot h_\mu) = \sum_{\nu \in N(\gamma)} h_\mu(P_\nu) b_\nu = (a \cdot h_\mu) = (e \cdot h_\mu) \text{ for } \mu = 1, \ldots, m$$

since $\{h_1, \ldots, h_m\}$ is a basis for $L(G)$. Thus the syndrome $(b - e \cdot h_\mu)$ is zero. Since the dual of C is defined as the set of all elements x in $\mathbb{F}_q^n$ which satisfy $(x \cdot \gamma) = 0$ for all $\gamma \in L(G)$, $b - e$ must be an element of this set. We have that

$$wt(b - e) \leq | \, N(\gamma) \, | \leq \deg(F) < \deg(G) - (2g - 2) = d^*$$

Since the minimum distance of C_Ω is greater than or equal to d^* we conclude that $b = e$ and $(e_\nu)_{\nu \in N(\gamma)}$ is an unique solution of (6.10). □

To sum up the discussion above we create the following algorithm:

Algorithm 2. Let the divisors D, G, F be given. Let $\{f_1, \ldots, f_l\}$, $\{g_1, \ldots, g_k\}$ and $\{h_1, \ldots, h_m\}$ be the basis for $L(F)$, $L(G-F)$ and $L(G)$ respectively. Let a be a received word in $\mathbb{F}_q^n$ such that $a = c + e$ for $c \in C_\Omega(D, G)$.

1. Find a solution $(\alpha_1, \ldots, \alpha_l) \neq 0$ of the linear system

$$\sum_{\lambda=1}^{l} (a \cdot f_\lambda g_\rho)\alpha_\lambda = 0 \quad \text{for} \quad \rho = 1, \ldots k \tag{6.11}$$

   Define $\gamma = \sum_{\lambda=1}^{l} \alpha_\lambda f_\lambda$. If there is no such solution for the system of equations above, a can not be decoded.

2. Compute the error positions by calculating the zeros of γ by computing

$$N(\gamma) = \{\nu \mid 1 \leq \nu \leq n \text{ and } \gamma(P_\nu) = 0\}$$

   This is done by evaluating $\gamma(P_\nu) = \sum_{\lambda=1}^{l} \alpha_\lambda f_\lambda(P_\nu)$ for $\nu = 1, \ldots, n$.

3. If

$$\sum_{\nu \in N(\gamma)} h_\mu(P_\nu) z_\nu = (a \cdot h_\mu) \quad \text{for} \quad \mu = 1, \ldots, m \tag{6.12}$$

   has a unique solution $(e_\nu)_{\nu \in N(\gamma)}$, we set $e = (e_1, \ldots, e_n)$ with $e_\nu = 0$ for all $\nu \notin N(\gamma)$. If there is no such solution of (7.3), we can not decode a.

4. Calculate $(c \cdot h_\mu)$ for $\mu = 1, \ldots, m$ to check if $c = a - e$ is an element of $\mathcal{C}_\Omega$ and if $wt(e) \leq t$. If this is the case, a is decoded to c. Otherwise we can not decode a.

This algorithm can be adapted to concider erasures as well as errors.

Next we will show an example of how to correct errors using the algorithm above.

Example 12. Again we look at the elliptic curve

$$\mathcal{E} : \{y^2 = x^3 + 9x + 4\} \cup \{(0:1:0)\}$$

over the finite field $\mathbb{F}_{13}$. By choosing divisors $G = 8P_\infty$ and $D = P_1 + P_2 + \ldots + P_{12}$ we can build the code $C_L(D, G)$ with the knowledge of the basis of the Riemann-Roch space of G which we recall from Example 9 is $\{1, x, x^2, x^4, y, y^2, xy, x^2y\}$. In Example 11 both the generator matrix A and parity check matrix H was calculated. Since we know that the parity check matrix of $C_L(D, G)$ is generator matrix for the dual of this code, we can choose a code word $c \in C_\Omega(D, G)$ among the rows of the matrix H.

Choose the code word

$$c = (5, 8, 2, 11, 1, 12, 12, 0, 0, 1, 0, 0, 0) \in C_\Omega(D, G)$$

which is to be sent over an unreliable channel. Assume that an error has occurred at the 7^{th} entry of c and the received word is

$$a = (5, 8, 2, 11, 1, 12, \underline{0}, 0, 0, 1, 0, 0, 0)$$

This implies that the error vector is $e = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$.

To continue we need to find a divisor F which satisfies equations (6.1), (6.2) and (6.3) as well as $\deg(F) \geq t + g = t + 1$ and $\deg(G - F) > t - 2g - 2 = t$. Considering these assumptions, let $F = 3P_{13}$. By the definition of the Riemann-Roch space $L(F)$ contains functions satisfying $\text{ord}_{P_{13}}(f) \geq -3$. This basis was also found in Example 9:

$$L(F) = \left\{1, \frac{1}{y}, \frac{1}{x-4}\right\}$$

Note, by choosing $F = 3P_{13}$ we are able to correct at most 2 errors. We also need the basis for $L(G - F)$, which was calculated to be

$$L(G - F) = \{y, y^2, xy, xy^2, x^2y\}$$

When all the divisors and their corresponding Riemann-Roch spaces are found, the error-locator function can be calculated. This is done by computing the syndromes $\sum_{\lambda=1}^{3}(a \cdot g_\rho f_\lambda)\alpha_\lambda$. This can be done by matrix multiplication. Let $\mathcal{F}$ be the matrix where entries are the basis elements of $L(F)$ evaluated in the points of the divisor D, $\mathcal{F} = [f_\lambda(P_i)]_{\lambda=1,i=1}^{3,12}$. Thus

$$\mathcal{F} = \begin{bmatrix} f_1(P_1) & f_1(P_2) \dots f_1(P_{12}) \\ f_2(P_1) & f_2(P_2) \dots f_2(P_{12}) \\ f_3(P_1) & f_3(P_2) \dots f_3(P_{12}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 6 & 1 & 12 & 7 & 6 & 1 & 10 & 12 & 3 & 7 & 6 \\ 3 & 3 & 4 & 4 & 6 & 6 & 7 & 10 & 7 & 10 & 2 & 2 \end{bmatrix}$$

In a similar way we define the matrix $\mathcal{G} - \mathcal{F}$ with entries $g_\rho(P_i)$ for $1 \leq \rho \leq 5$ and $1 \leq i \leq 12$:

$$(\mathcal{G} - \mathcal{F}) = \begin{bmatrix} g_1(P_1) & g_1(P_2) & \dots & g_1(P_{12}) \\ g_2(P_1) & g_2(P_2) & \dots & g_2(P_{12}) \\ \vdots & & & \vdots \\ g_5(P_1) & g_5(P_2) & \dots & g_5(P_{12}) \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 11 & 1 & 12 & 2 & 11 & 1 & 4 & 12 & 9 & 2 & 11 \\ 4 & 4 & 1 & 1 & 4 & 4 & 1 & 3 & 1 & 3 & 4 & 4 \\ 0 & 0 & 1 & 12 & 4 & 9 & 6 & 6 & 7 & 7 & 9 & 4 \\ 0 & 0 & 1 & 1 & 8 & 8 & 6 & 11 & 6 & 11 & 5 & 5 \\ 0 & 0 & 1 & 12 & 8 & 5 & 10 & 9 & 3 & 4 & 8 & 5 \end{bmatrix}$$

Let d be a diagonal matrix with the received word a as the entries on the diagonal and $\alpha = [\alpha_1, \alpha_2, \alpha_3]$. Thus we obtain

$$\sum_{\rho=1}^{5}\sum_{\lambda=1}^{3}(a \cdot f_\lambda g_\rho)\alpha_\lambda = (\mathcal{G} - \mathcal{F})(\mathcal{F}d)^T\alpha^T = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 1 & 7 \\ 6 & 6 & 9 \\ 6 & 6 & 9 \\ 6 & 6 & 3 \\ 10 & 10 & 3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

Hence we have a system of equations with 3 unknowns and 5 equations. By choosing 3 linearly independent equations we obtain that $\alpha_1 = 12$, $\alpha_2 = 1$ and $\alpha_3 = 0$. When we have found a solution to the syndromes, we calculate the error locator function $\gamma = 12 + \frac{1}{y}$. Since $\gamma(P_3) = \gamma(P_4) = \gamma(P_7) = \gamma(P_9)$. This implies that

$$N(\gamma) = \{3, 4, 7, 9\}$$

Hence we can expect an error to have occurred at place 3, 4, 7 and 9 in c.

Next we calculate the error values $(e_\nu)_{\nu \in N(\gamma)}$.This is done by solving the system

$$\sum_{\nu \in N(\gamma)} h_\mu(P_\nu)e_\nu = (a \cdot h_\mu) \text{ for } h_\mu \in L(G) \text{ and } \mu = 1, \ldots, 8$$

We get 8 equations with 4 unknowns. By choosing the following 4 linearly independent equations

$$\begin{aligned} e_3 + e_4 + e_8 + e_{10} &= 1 \\ e_3 + e_4 + 6x_8 + 6x_{10} &= 6 \\ e_3 + 12e_4 + e_8 + 12e_{10} &= 4 \\ e_3 + 12e_4 + 6e_8 + 7e_{10} &= 9 \end{aligned}$$

we obtain $(e_{\nu \in N(\gamma)}) = (0, 0, 1, 0)$ which yields error vector $e = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$. By calculating $a - e$ we get $(5, 8, 2, 11, 1, 12, 12, 0, 1, 0, 0, 0)$ which is the code word c we were searching for.

Next we show to what extend the algorithm is able to correct errors and erasures. Let $a = c + e + r$ be a received word with $e = (e_1, \ldots, e_t) \in \mathbb{F}_q^n$ the vector of errors with Hamming weight t and $r = (r_1, \ldots, r_\tau) \in \mathbb{F}_q^n$ the vector of erasures of Hamming weight τ. Considering this, denote $E = \{E_1, \ldots, E_t\}$ to be the set of error locators and $R = \{R_1, \ldots, R_\tau\}$ to be the set of erasure locators. Let $\{k_1, \ldots, k_s\}$ be the basis for $L(F - \sum R_j)$.

Theorem 6. *Let $C_\Omega(D, G)$ be a algebraic geometry code associated with the curve χ of genus g such that $\deg(G) = a$ and $2g - 2 < a \leq n + g - 1$. Let t and τ be nonnegative*

integers satisfying

$$l(F) > t + \tau$$
$$a - b > t + 2g - 2$$

for a divisor F of degree $b \leq a$. Then the basic algorithm can correct t errors and τ erasures.

To prove this theorem, we rewrite it as the following proposition and give a proof of this instead.

Proposition 5. *For divisors D, G and F of the curve χ and the code $C_\Omega(D, G)$:*
(a) Assume that $l(F') > t + \tau$ holds, then there is at least one non-trivial solution of $\sum_{i=1}^{s}(a, k_i g_j)x_i = 0$ for $j = 1, \ldots, k$.
(b) Assume that $a-b > t+2g-2$ holds, then for any solution $(y_1, \ldots, y_s)$ of $\sum_{i=1}^{s}(a, k_i g_j)x_i = 0$ for $j = 1, \ldots, k$ we have $k_y(E_i) = 0$ for any $i = 1, \ldots, t$ where $k_y = y_1 k_1 + \ldots + y_s k_s \in L(F - \sum R_j)$. This means that k_y has all the errors and erasure locators among its zeros.

Proof. For the first part, let $a = e + r + c$ as before. From the starting assumption, $l(F - E_1 - \ldots - E_t - R_1 - \ldots - R_\tau) > 0$. Thus there exists at least one non-zero element, g in the space $L(F - \sum R_j)$ such that $g(E_i) = 0$ for $i = 1, \ldots, t$. Let $y_1 k_1 + \ldots + y_s k_s$ be the decomposition of k with respect to the basis of $L(F - \sum R_i)$. To proceed we claim the following

Claim. *$y = (y_1, \ldots, y_s)$ is a solution of $\sum_{i=1}^{s}(a \cdot k_i g_j)x_i = 0$*

Since $(c \cdot k_i g_j)$ is zero and $k_i(R_j)$

$$\sum_{i=1}^{s}(a \cdot k_i g_j)y_i = \sum_{i=1}^{s}(e + r \cdot k_i g_j)y_i = \sum_{i=1}^{s}\sum_{m=1}^{t} e_m k_i(E_m) g_j(E_m) y_i$$
$$= \sum_{m=1}^{t} e_m g_j(E_m) k(E_m) = 0$$

Thus $y = (y_1, \ldots, y_s)$ is a solution of $\sum_{i=1}^{s}(a \cdot k_i g_j)x_i = 0$.

For the last part of the theorem note that for a canonical divisor W the Riemann-Roch theorem implies

$$l(G - F - \sum E_i) - l(W - (G - F - \sum E_i)) = \deg(G - F - \sum E_i) + 1 - g$$

Then by the assumption $a - b > t + 2g - 2$

$$\deg(G - F - \sum E_i) = a - b - t > 2g - 2$$

Since $\deg W = 2g-2$ it follows that $l(W-(G-F-\sum E_i))=0$ and $l(G-F-\sum E_i) = a-b-t+1-g$. In a similar way, since $\deg(G-F) = a-b > 2g-2+t > 2g-2$, $l(G-F) = a-b-g+1$.

Define $E = \{E_1, \ldots, E_t\}$ and consider the evaluation map

$$\begin{aligned} ev_E : L(G-F) &\to \mathbb{F}_q^t \\ z \in L(G-F) &\mapsto (z(E_1), \ldots, z(E_t)) \end{aligned}$$

It is easy to show that the kernel of this map is $L(G-F-\sum E_i)$ and it can also be shown that it is surjective.

Fix any $j = \{1, \ldots, n\}$. Then it is possible to find a element g in $L(G-F)$ so that $g(E_j) = 1$ and $g(E_m) = 0$ for $E_m \neq E_j$. In this case the linear combination of equations is

$$\sum_{i=1}^{s} (a \cdot k_i g) x_i = 0$$

Let $(y_1, \ldots, y_s)$ be a non-trivial solution to the above set of equations. We then obtain

$$0 = \sum_{i=1}^{s} (a \cdot k_i g) y_i = \sum_{i=1}^{s} \sum_{m=1}^{t} e_m k_i(E_m) g(E_m) y_i = e_j k_y(E_j)$$

Since $e_j \neq 0$, $k_y(E_j)$ must be equal to zero. □

The basic algorithm can correct up to $\lfloor (d^* - 1 - g)/2 \rfloor$. Hence it fails to correct all errors (included erasures) of weight $(d^*-1)/2$. This follows from the fact that the set of zeros of an error-locator function is not assumed to be equal to the set of error positions. When a error-locator function has zeros at t prescribed points, in general it also have g other zeros. Hence the deficiency of the algorithm depends on the genus of the curve. By studying Algorithm 2 the complexity of the basic algorithm is $\mathcal{O}(n^3)$ where n is the length of the code.

7 The Modified Decoding Algorithm

In this section some modifications are applied to the basic algorithm to improve the number of errors corrected. By applying the basic algorithm with an increasing sequence of divisors $F_1, \ldots, F_s$ we get the so called modified decoding algorithm. This algorithm has a higher correcting rate than the basic decoding algorithm which can correct up to $[(d^* - 1 - g)/2]$ errors but has a slower running time.

As for the basic algorithm we look at divisors $D = P_1 + P_2 + \ldots + P_n$ and G of a rational curve of genus $g \geq 1$ with the assumption that $\mathrm{supp}(G) \cap D = \emptyset$. As an additional assumption on G we demand that this divisor is a multiple of an effective divisor H, i.

e. $G = a_1 H$. Say H is a divisor of degree h, then $\deg(G) = a_1 h := a$. In the modified basic decoding algorithm F is chosen to be the set $\{iH\}_{i=1}^{b_1}$ where b_1 is the least integer for which the set of equations in (6.11) has a non-trivial solution.

Remark. By demanding that the divisor G is a multiple of an effective divisor there is a restriction on the codes which the modified algorithm can decode. Hence not all algebraic geometry codes can be decoded by the modified algorithm.

Let $\{f_{b1}, \ldots, f_{bl_b}\}$ be the basis of $L(G) = L(a_1 H)$, $\{g_{b1}, \ldots, g_{bk_b}\}$ the basis of $L(G-F) = L((a_1 - b)H)$ and $\{h_1, \ldots, h_m\}$ the basis for $L(G)$. For simplicity we define the following matrices $\mathcal{F} =\mid f_{bi}(P_\nu) \mid_{i=1,\nu=1}^{l_b,n}$, $\mathcal{G} =\mid g_{bi}(P_\nu) \mid_{i=1,\nu=1}^{k_b,n}$ and $\mathcal{H} =\mid h_i(P_\nu) \mid_{i=1,\nu=1}^{m,n}$.

As in the discussion for the basic decoding algorithm, $f_{b\lambda} g_{g\rho}$ is an element in $L(G) = L(a_1 H)$ for $1 \leq \lambda \leq l_b$ and $1 \leq \rho \leq k_b$ and hence $ev_D(f_{b\lambda} g_{g\rho})$ is in $C_L(D, G)$ which implies that $ev_D(f_{b\lambda} g_{g\rho}) x = 0$ for x in the dual of $C_L(D, G)$. With this information, the modified algorithm is constructed as follows

Algorithm 3 (Modified Decoding Algorithm)**.** Let $a = c + e$ be a received word so that $c \in C_\Omega(D, G)$ and $b = 1$.

1. Find a non-trivial solution $\alpha = (\alpha_1, \ldots, \alpha_{l_b})$ of

$$\sum_{\lambda=1}^{l_b} (a \cdot f_\lambda g_\rho)\alpha_\lambda = \sum_{\lambda=1}^{l_b} \sum_{\nu=1}^{n} a_\nu f_\lambda(P_\nu) g_\rho(P_\nu) \alpha_\lambda = 0 \quad \text{for} \quad \rho = 1, \ldots, k_b \tag{7.1}$$

   If no such solution exist update b by $b = b + 1$ and return to Step 1. If there exists a nontrivial solution of the system put $F = bH$ and proceed to Step 2.

2. Define

$$\gamma = \sum_{\lambda=1}^{l_b} \alpha_\lambda f_\lambda \quad \text{for} \quad f_\lambda \in L(bH) \tag{7.2}$$

   and calculate the zeros of γ by finding the $N(\gamma) = \{\nu \mid 1 \leq \nu \leq n \quad \text{and} \quad \gamma(P_\nu = 0)\}$.

3. If

$$\sum_{\nu \in N(\gamma)} h_\mu(P_\nu) e_\nu = (a \cdot h_\mu) \quad \text{for} \quad \mu = 1, \ldots, m \tag{7.3}$$

   has a unique solution $(e_\nu)_{\nu \in N(\gamma)}$, we set $e = (e_1, \ldots, e_n)$ with $e_\nu = 0$ for all $\nu \notin N(\gamma)$. If there is no such solution of (7.3), we can not decode a.

4. Calculate $(c \cdot h_\mu)$ for $\mu = 1, \ldots, m$ to check if $c = a - e$ is an element of $\mathcal{C}_\Omega$ and if $wt(e) \leq t$. If this is the case, a is decoded to c. Otherwise we can not decode a.

In the next example we show how the modified decoding algorithm is used to decode algebraic geometry codes.

Example 13. Consider the elliptic curve $\mathcal{E} : \{y^2 = x^3 + 9x + 4\} \cup \{(0 : 1 : 0)\}$ over $\mathbb{F}_{13}$. Let $D = P_1 + P_2 + \ldots + P_{13}$ and $G = 8P_\infty$ and $C_\Omega(D, G)$ has generator matrix H as in Example 11. Choose $a = (5, 8, 2, 11, 1, 12, 0, 0, 0, 1, 0, 0, 0)$ to be a received word equal to $c + e$ for some c in $C_\Omega(D, G)$. Let $F = \{iP_\infty\}_{i=1}^{b}$ and $\{1, x, x^2, x^4, y, y^2, xy, x^2y\}$ be the basis for $L(G)$.

First we search for a b which give a non-trivial solution to the system in (7.1). Let $b = 1$. Then $F = P_\infty$ and $L(P_\infty)$ only contain the constant functions. It follows that $L(G - F) = L(7P_\infty) = \{1, x, x^2, y, y^2, xy, x^2y\}$. If d is the diagonal matrix with the vector a on its diagonal we get

$$\sum_{\lambda=1}^{1}\sum_{\nu=1}^{8} a_\nu f_\lambda(P_\nu) g_\rho(P_\nu)\alpha_\lambda = \mathcal{G}(\mathcal{F}d)^T\alpha^T = 0 \quad \text{for} \quad \rho = 1, \ldots, 7$$

This gives 7 equations with one unknown, $\alpha = (\alpha_1) = 0$. Since the system only has a non-trivial solution, b is updated to $b + 1$.

For $b = 2$, $F = 2P_\infty$ and $G - F = 6P_\infty$. In this case $L(F)$ and $L(G-F)$ has basis $\{1, x\}$ and $\{1, x, x^2, y, y^2, xy\}$ respectively. Also this case gives the trivial solution for equation (7.1). Again we update b by $b + 1$ and try to find a non-trivial solution to the syndromes.

When $b = 3$, the divisors F and $G - F$ are $3P_\infty$ and $5P_\infty$ respectively. The Riemann-Roch spaces of these divisors are

$$L(F) = \{1, x, y\} \quad \text{and} \quad L(G - F) = \{1, x, x^2, y, xy\}$$

The syndromes $\sum_{\lambda=1}^{l_b}(a \cdot f_\lambda g_\rho)\alpha_\lambda$ are

$$\mathcal{G}(\mathcal{F}d)^T\alpha^T = \begin{bmatrix} 1 & 6 & 1 \\ 6 & 10 & 6 \\ 10 & 8 & 10 \\ 1 & 6 & 1 \\ 6 & 10 & 6 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

This system of equations gives a non-trivial solution $\alpha = (12, 0, 1)$. Thus we can calculate the error locator function $\gamma = \sum_{\lambda=1}^{3}\alpha_\lambda f_\lambda = 12 + y$. This implies that $N(\gamma) = \{3, 7\}$ which means that an error may have occurred in entry 3 and 7 of a.

To find the error vector we calculate $\sum_{\nu\in N(\gamma)} h_\mu(P_\nu)e_\nu = (a \cdot h_\mu) \quad \text{for} \quad \mu = 1, \ldots, 5.$ We obtain that $e_3 = 0$ and $e_7 = 1$ which implies that the error vector $e = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$. Thus

$$c = a - e = (5, 8, 2, 11, 1, 12, 12, 0, 0, 1, 0, 0, 0)$$

The code word c is an element of $C_\Omega(D, G)$ since $(c \cdot h_\mu) = 0$ for $\mu = 1, \ldots, 8$. Hence a is decoded to c.

The next theorem shows that the modified algorithm is an improvement on the basic algorithm:

Theorem 7. *Algorithm 3 corrects up to* $[(d(C)-1)/2] - S(H)$*, where* $S(H)$ *is the characteristic of* H *defined by*

$$S(H) = \max_{i \in \mathbb{Z}} \{[(ih + h + 1)/2 - l(ih)]\} \tag{7.4}$$

Proof. Consider a system of linear equations as

$$\sum_{i=1}^{s} (a \cdot k_i g_j) x_i = 0 \tag{7.5}$$

and consider the code $\mathcal{C}_\Omega(D, G)$ where $D = P_1 + \ldots + P_n$ and G are divisors of a rational curve χ. Consider another divisor F of *chi* so that $F = H, 2H, \ldots, bH$ where b is the least integer so that the set of the equations in (7.5) has a solution not equal to zero. From Proposition 5 we can derive that $l((b-1)H - \sum E_i) = 0$.

We claim the following

Claim. *The divisor* $(a_1 - b)H - \sum E_i$ *is not special.*

Assume that $(a_1 - b)H - \sum E_i$ is special. This means that the divisor is linearly equivalent to $W - J$ for a canonical divisor W and a divisor $J \geq 0$ with $\deg(J) = 2g - 2 - a_1 h + bh + t$ of χ. In this case

$$(a_1 - b)H - \sum E_i \sim W - J$$

which by a simple rearrangement can be written as

$$-bH - \sum E_i \sim W - J - a_1 H$$

Hence

$$bH - \sum E_i \sim W - J - a_1 + 2bH$$

By gathering terms we obtain that

$$(b-1)H - \sum E_i \sim W - J - (a_1 - 2b + 1)H$$

Hence $l(W - J - (a_1 - 2b + 1)H) = 0$ and $l(W - J - (a_1 - 2b + 1)H) \leq \deg(J)$. On the contrary, by the Riemann-Roch theorem

$$l((a_1 - 2b + 1)H) - l(W - (a_1 - 2b + 1)H) = \deg((a_1 - 2b + 1)H) - g + 1$$

Hence $l(W-(a_1-2b+1)H) = l((a_1-2b+1)H) - a_1 + 2bh - h + g - 1$. Then, from the definition of the characteristic of H we obtain

$$\begin{aligned} l((a_1-2b+1)H) &\geq [((a_1-2b+1)h+h+1)/2] - S(H) \\ &= [(a_1h-2g+1)/2] + g - bh + h - S(H) \end{aligned}$$

Therefore $l(W-(a_1-2b+1)H) \geq [(d^*-1)/2] + 2g - 1 - a_1h + bh - S(H) \geq \deg(J) + 1$ which contradicts the fact that the divisor $(a_1-b)H - \sum E_i$ is special and hence the claim is proven.

This means that $(a_1-b)H$ is also non-special. Thus

$$l((a_1-b)H) - l((a_1-b)H - \sum E_i) = 1$$

It follows from Proposition 5 part (b) that for any non-zero solution $y = (y_1, \ldots, y_{l_b})$, $k_y(E_i) = 0$ for all $i = 1, \ldots, t$. Here $k_y = y_1k_1 + \ldots + y_ik_{l_b} \in L(bH)$. This implies that all the error locations are among the zeros of k_y.

To proceed, let $\{Q_1, \ldots, Q_u\}$ be the set of the zeros of k_y in D. For $i = 1, \ldots, t$ we may assume that $Q_i = E_i$. The values of the errors may be found from

$$(a \cdot h_i) = \sum_{i=1}^{u} h_j(Q_i)z_i \quad \text{for} \quad j = 1, \ldots, m$$

We need to show that $(e_1, \ldots, e_t, 0, \ldots, 0)$ is a solution of the set of linear equations above. By following the same steps as in proof of the proposition which shows how to find the error values, we see that as long as $\deg(bH) = bh$ is less than the designed distance, $(e_1, \ldots, e_t, 0, \ldots, 0)$ is the solution to our system. Thus, since $l((b-1)H - \sum E_i)$ is zero, $l((b-1)H) \leq t$. By the definition of $S(H)$, $[(bh+1)/2] \leq l((b-1)H) + S(H)$ and it follows that

$$[(bh+1)/2] \leq t + S(H) \leq [(d^*-1)/2]$$

hence $\deg(bH) = bh \leq d^*$.

Hence we can conclude that the modified basic algorithm corrects up to $[(d^*-1)/2] - S(H)$ errors. □

From the discussion above we have proved that by modifying the basic algorithm by adding some extra restrictions to the divisors G and F we can correct up to $[(d*-1)/2] - S(H)$ errors where $S(H)$ is the characteristic of a divisor H. There are two cases where $S(H) = 0$ and the algorithm can correct up to half the designed minimum distance. If $\deg H = h$ and the code is build from the curve χ, $S(H)$ is zero if

1. χ is an elliptic curve, and h is equal to 1 or 2.

2. χ is a hyperelliptic curve, and the divisor H is a Weierstrass point of a hyperelliptic divisor.

For proof of this fact see [21]. Here you can also find the proof of the fact that $S(H) \leq g/2$ which is essential for the modified basic algorithm to correct more errors and erasures than the basic algorithm. This algorithm requires at most $\mathcal{O}(n^4)$ elementary operations when n is the length of the code.

8 The Extended Modified Decoding Algorithm

When modifying the basic decoding algorithm, we get an algorithm which restricts to a certain class of codes. In this section we search for an algorithm which can decode all types of codes. The modified algorithm is extended by applying the theory of special divisors in the decoding process, i. e. we apply the basic algorithm to a sequence of special divisors. This algorithm shows a defect on the curve connected to the code instead of the code directly. In other words, the success of the algorithm depends on the curve which the code is build on.

In this section the code $C_\Omega(D, G)$ is considered, where $D = P_1 + P_2 + \ldots + P_n$ and G are divisors on the same curve of genus g such that $\text{supp}(G) \cap D = \emptyset$. Let $a = c + e$ be a received word where c is a code word of $C_\Omega(D, G)$ and e is an error vector. The set of error positions are denoted $I = \{\nu \mid 1 \leq \nu \leq n, e_\nu \neq 0\}$. The zeros of the error locator function γ are contained in the set $N(\gamma) = \{\nu \mid 1 \leq \nu \leq n, \gamma(P_\nu) = 0\}$. Thus the error divisor is $P = \sum_{\nu \in N(\gamma)} P_\nu$. The next theorem shows the existence of the extended modified algorithm

Theorem 8 (The Extended Modified Algorithm)**.** *Let D and G be divisors as above. Let $C = C_\Omega(D, G)$ be a residue code, defined on a curve χ of genus g. Let the designed minimum distance be odd and defined by $d^* = 2e + 1$. Let $\varepsilon = \{E_0, E_1, \ldots, E_{2g-2}\}$ be a set of special divisors on the curve χ as in Definition 10 and let ε_0 be the subset of ε containing divisors of even degree, say*

$$\varepsilon_0 = \{E_0, E_1, \ldots, E_{g-1}\} \text{ and } \sigma_0 = \sigma(\varepsilon_0) \tag{8.1}$$

where $\sigma_0 = \sigma(\varepsilon_0)$ is the Clifford's defect of the set ε_0. Then

$$\deg(E_i) = 2g - 2 - 2i \quad \text{and} \quad \deg(E_i)/2 - (l(E_i) - 1) \leq \sigma_0 \tag{8.2}$$

for $i = 0, 1, \ldots, g - 1$.

Let $\mathcal{F} = F_0, F_1, \ldots, F_g$ be a set of divisors on our curve satisfying

$$\begin{aligned} &\deg(F_0) = e \\ &F_i \cap D = \emptyset \quad \text{for} \quad = 0, 1, \ldots, g \\ &F_i \sim G - F_{i-1} - E_{i-1} \quad \text{for} \quad i = 1, 2, \ldots, g \end{aligned} \tag{8.3}$$

Then a received word with $t \leq (d^* - 1)/2 - \sigma_0$ errors, can be corrected by successive applications of the basic algorithm with $F = F_i$ of lowest possible degree so that both $L(F - P) \neq 0$ and $\Omega(G - F - P) = 0$ are satisfied.

The fact that $L(F - P) \neq 0$ and $\Omega(G - F - P) = 0$ are satisfied ensures that a non-zero error locator function γ can be found. But these two conditions conflict since $L(F-P) \neq 0$ holds for sufficiently large degree of F while $\Omega(G - F - P) = 0$ hold for a divisor F of sufficiently low degree. The next lemma will prove helpful when we want to show that there is possible to choose a set of divisors such that the conditions above are satisfied for at least one element of the set.

Lemma 3. *Let G, F and $P = \sum_{\nu \in N(\gamma)} P_\nu$ be divisors on a curve with genus g. Assume E and F^* are divisors which satisfy*

$$l(E) \geq g - \deg(F - P) \quad \textit{and} \quad F^* \sim G - F - E \tag{8.4}$$

Then

$$L(F - P) = 0 \quad \textit{implies that} \quad \Omega(G - F^* - P) = 0 \tag{8.5}$$

Proof. Let χ be a given curve of genus g with divisors G, F^* and $P = \sum_{\nu \in N(\gamma)} P_\nu$. Assume that $\Omega(G - F^* - P) \neq 0$. Let w be a non-zero element in $\Omega(G - F^* - P)$. When this is the case, we will prove that $L(F - P) \neq 0$.

Let $(w) = G - F^* - P - E^*$ for an integral divisor E^*. Since F^* is in the same equivalence class as $G - F - E$ we obtain

$$(w) \sim G - (G - F - E) - P + E^* = F + E - P + E^*$$

Since w is defined as a non-zero element of $\Omega(G - F^* - P)$, (w) is a canonical divisor and equivalent to all other canonical divisors on the curve χ. When W is an element of this class we obtain that

$$F - P \sim W - E - E^*$$

Since $E^* \geq 0$ by assumption, it suffices to prove that $\deg(E^*) < l(W-E)$ for $L(F-P) \neq 0$. This follows from the fact that

$$W - E \sim G - F^* - P - E \sim G - (G - F - E) - P - E = F - P$$

where $W \sim G - F^* - P$ is used in the first equivalence relation and $F^* \sim G - F - E$ in the second. It is a known fact that when two divisors are in the same equivalence class then the corresponding Riemann-Roch spaces are isomorphic as vector spaces. Since $L(W - E)$ is isomorphic to $L(F - P)$ the dimension of these Riemann-Roch spaces are the same. Hence $l(W - E) = l(F - P)$. For $L(F - P)$ to be non-zero, the dimension of the Riemann-Roch

space has to be greater than zero, i.e. $l(W-E) = l(F-P) > 0$. This is definitely satisfied if $l(W-E) > \deg(E^*)$, since E^* is defined to be an effective divisor, which implies that $\deg(E^*)$ is greater than or equal to zero.

By assumption

$$l(E) \leq g - \deg(F-P)$$

By substituting $F - P \sim W - E - E^*$ in the equation above we obtain

$$\begin{aligned} l(E) + 1 > l(E) &\geq g - \deg(W - E - E^*) = g - \deg(W) + \deg(E) + \deg(E^*) \\ &= g - 2g + 2 + \deg(E) + \deg(E^*) \end{aligned}$$

In the last step we have used that the degree of a canonical divisor is $2g-2$ and we obtain that

$$\deg(E^*) < l(E) + g - 1 - \deg(E)$$

From the Riemann-Roch theorem we have the following relationship between E and K:

$$l(E) = l(K-E) + \deg(E) + 1 - g$$

Thus

$$\deg(E^*) < (l(W-E) + \deg(E) + 1 - g) + g - 1 - \deg(E) = l(W-E) \tag{8.6}$$

This implies that $\deg(E^*) < l(F-P)$. Hence we have proven that when $\Omega(G - F^* - P)$ is zero then $L(F-P)$ is zero. □

The next lemma gives a bound on the error correction t.

Lemma 4. *Let $C = C_\Omega(D, G)$ be a code as before, hence it has designed minimum distance $d^* = \deg(G-W)$. Also let $P = \sum_{\nu \in N(\gamma)} P_\nu$ so that $t = \deg(P)$. Then the inequality $l(E) \geq g - \deg(F-P)$ from the previous lemma is equivalent to*

$$t \leq \left(\frac{d^*-1}{2}\right) + (l(E)-1) - \left(\frac{\deg(E)}{2}\right) - \left(\frac{\deg(F^*-F)-1}{2}\right) \tag{8.7}$$

Proof. We start by multiplying equation (8.7) with 2 to obtain

$$2t \leq 2 \cdot \left(\frac{d^*-1}{2}\right) + 2 \cdot (l(E)-1) - 2 \cdot \left(\frac{\deg(E)}{2}\right) - 2 \cdot \left(\frac{\deg(F^*-F)-1}{2}\right)$$

The fact that $t = \deg(P)$ and $d^* = \deg(G-W)$ implies that

$$2\deg(P) \leq \deg(G-W) - 1 + 2(l(E)-1) - \deg(E) - \deg(F^*-F) + 1$$

By applying the equivalence relation $F^* \sim G - F - E$ we obtain that

$$\begin{aligned} 2\deg(P) &\leq \deg(G - W) + 2(l(E) - 1) - \deg(E) - \deg(G - 2F - E) \\ &= -\deg(W) + 2(l(E) - 1) - \deg(G) + \deg(2F) \end{aligned}$$

Which again implies that $\deg(W) - 2\deg(F - P) \leq 2(l(E) - 1)$ and thus

$$(g - 1) - \deg(F - P) \leq l(E) - 1 \quad \text{which implies} \quad l(E) \leq g - \deg(F - P)$$

□

When we now have established the facts above we are ready to prove the existence of the extended modified algorithm.

Proof. (Extended Modified Algorithm) Let P be the divisor of all error locations. When e_P is the set of error positions , $P = \sum_{e_P \neq 0} P$. This divisors has degree t which is the weight of the received word. Assume that W is a canonical divisor of the curve. Consider divisors G, F_{i-1} , E_{i-1} and F_{i-1} of the same curve satisfying

$$\begin{aligned} &\deg(F_{i+1} - F_i) = 1 \\ &\frac{\deg(E_i)}{2} - (l(E_i) - 1) \leq \sigma_0 \\ &\deg(P) \leq e - \sigma_0 \end{aligned} \tag{8.8}$$

From our assumption we have $F_i \sim G - F_{i-1} - E_{i-1}$ where $F_i \cap D = \emptyset$ for $i = 1, 2, \ldots, g$. Hence

$$\deg(F_i + E_{i-1}) = \deg((G - F_{i-1} - E_{i-1}) + E_{i-1}) = \deg(G - F_{i-1})$$

By $F_{i-1} \sim G - F_{i-2} - E_{i-2}$

$$\deg(F_i + E_{i-1}) = \deg((F_{i-1} + F_{i-2} + E_{i-2}) - F_{i-1}) = \deg(F_{i-2} + E_{i-2})$$

By moving terms in the above assumption and applying that $\deg(E_i) = 2g - 2 - 2i$

$$\begin{aligned} \deg(F_i - F_{i-2}) &= \deg((G - F_{i-1} - E_{i-1}) - (G - F_{i-1} - E_{i-2})) \\ &= \deg(E_{i-2} - E_{i-1}) = (2g - 2 - 2(i - 2)) - (2g - 2 - 2(i - 1)) \\ &= (-2i + 4) + (2i - 2) = 2 \end{aligned}$$

Since $\deg(F_0) = e$, $\deg(F_1) = e + 1$ and $\deg(F_i - F_{i-2}) = 2$ we obtain the following by applying a simple induction algorithm

$$\deg(F_i) = e + i \quad \text{for} \quad i = 0, 1, \ldots, g.$$

By assumption $\deg(P) = t \leq e - \sigma_0$ for $\sigma_0 \geq 0$ where σ_0 is the same as in Definition 10. Since $\deg(F_0) = e$, $\deg(P) \leq e - \sigma_0$, $d^* = \deg(G - W)$ and $d^* = 2e + 1$ it follows that

$$\begin{aligned}\deg(G - F_0 - P) &\geq (\deg(W) + 2e + 1) + (-e) - (e - \sigma_0) \\ &= \deg(W) + 1 + \sigma_0 \geq \deg(W) + 1\end{aligned}$$

Hence $\deg(G - F_0 - P) > \deg(W)$ which implies that $G - F_0 - P$ can not be a canonical divisor. Hence the corresponding set of rational differential forms must be zero, i. e. $\Omega(G - F_0 - P) = 0$.

To finish this proof, our next step is to prove that $L(F_i - P) = 0$ implies that $\Omega(G - F_{i+1} - P) = 0$ for $i = 0, 1, \ldots, g-1$. Since $F_i \sim G - F_{i-1} - E_{i-1}$ and $\deg(E_i)/2 - (l(E_i) - 1) \leq \sigma_0$ by assumption, we obtain

$$l(E_i) \geq \deg(E_i)/2 + 1 - \sigma_0$$

By applying that $\deg(E_i) = 2g - 2 - 2i$ and $\deg(P) \leq e - \sigma_0$ in the equation above we get

$$\begin{aligned}\deg(E_i)/2 + 1 - \sigma_0 &= g - i - \sigma_0 \geq g + \deg(P) - (i + e) \\ &= g + \deg(P) - \deg(F_i) = g - \deg(F_i - P)\end{aligned}$$

Thus $l(E_i) \geq g - \deg(F_i - P)$ and all conditions in Lemma 3 are satisfied. Hence $L(F_i - P) = 0$ implies that $\Omega(G - F_{i+1} - P) = 0$. To prove that this only holds for values of i less than g consider F_g and note that

$$\deg(F_g - P) \geq (e - g) - (e - \sigma_0) \geq g$$

Since the genus is a non-negative integer, the Riemann-Roch space of the divisor $F_g - P$ can not be zero. This implies that there is possible to find a set of special divisors where at least one divisor satisfy that $\Omega(G - F_{i+1} - P) = 0$ and $L(F - P) \neq 0$, i. e the divisor F_g.

Considering that the minimum distance is $2e + 1$ we have that $e = (d^* - 1)/2$ and

$$t = \deg(P) \leq e - \sigma_0 = \frac{d^* - 1}{2} - \sigma_0$$

holds for all choices of F. □

Consider a code $C_\Omega(D, G)$ of even designed minimum distance $d^* = 2e + 2$ and the set of special divisors of odd degree $\varepsilon_1 = \{E_1, \ldots E_{g-1}\}$ instead of ε_0 so that $\sigma_1 = \sigma(\varepsilon_1)$. Let the divisors in the set ε satisfy $\deg(E_i) = 2g - 1 - 2i$ and $\deg(E_i)/2 - (l(Ei) - 1)$ for $i = 1, 2, \ldots, g - 1$. Let $\mathcal{F}$ be a set of divisors which satisfies the same conditions as in Theorem 8. With a similar procedure as in the proof of the existence of the extended modified algorithm for codes with odd designed minimum distance, we can prove that in the case of even designed minimum distance, the extended modified algorithm corrects up to $((d^* - 1)/2) - \sigma_1$ errors.

Remark. It can be proven that $0 \leq \sigma \leq (g-1)/2$ for all curves of genus g greater than or equal to 1. This implies that the extended modified decoding algorithm has a higher correction rate than the basic decoding algorithm. (For a proof see [20].)

We summarize the extended modified algorithm in

Algorithm 4 (Extended Modified Algorithm)**.** Let $a \in \mathbb{F}_q^n$ be given.

1. Set $i = 1$ and let $F = F_i$.
2. Calculate $L(F - P)$. If this is equal to zero, update $i = i + 1$ and set $F = F_i$ and go to step 2. Otherwise proceed to step 3.
3. Put $F = F_i$ and apply Algorithm 2.

The success of the extended modified decoding algorithm depends on the curve which the codes is build from. Let χ be a curve. For elliptic curves $\sigma(\chi) = 0$ with $\varepsilon = \{0\}$. In this case, decoding up to half the designed minimum distance is possible. The same hold for curves with $\sigma(\chi) = 1/2$. This is the case when $\sigma_0 = 0$ and $\sigma_1 = 1/2$. The next proposition states for which curves the extended decoding algorithm decodes up to the designed error correction bound $t = \lfloor (d^* - 1)/g \rfloor$.

Proposition 6. *A curve χ satisfying $\sigma(\chi) \leq 1/2$, is either a curve of genus genus zero or one or a hyperelliptic curve.*

Hence the codes which allow decoding up to half the designed minimum distance are curves build from lines, conics, singular cubics, elliptic curves and hyperelliptic curves. Thus for rational, elliptic and hyperelliptic curves the extended algorithm is t error correcting for $t \leq \lfloor (d^* - 1)/2 \rfloor$. For a proof of this proposition see [20]. There are also certain curves with $\sigma(\chi) = 1$ which decode up to half the designed minimum distance, but these will not be mention in this paper. In [20] Clifford's defect is calculated for several other curves.

The next theorem tells us how fast the extended modified algorithm is.

Theorem 9. *When n is the length of the code, the complexity of Algorithm 4 is at most $\mathcal{O}(n^3)$.*

9 List Decoding of Algebraic Geometry Codes

When decoding algebraic geometry codes, unique decoding is usually impossible as long as the number of errors is larger than $(d-1)/2$, where d is the code's minimum distance. In this section a method for decoding algebraic geometry codes even when the errors exceed this limit is derived. This is the list decoding algorithm for AG codes. It requires as input a received word a and an error bound e. The output is a list of all code words $c_1, \ldots, c_m \in C$

which differ from a in at most t coordinates. This list decoding algorithm is also known as the Sudan-Guruswami algorithm.

Consider an $[n, k, d]$-code $C = C_L(D, G)$ where $D = P_1 + \ldots + P_n$ and G are divisors of a curve χ of genus g as in previous sections. Let K be the curve's function field. Assume that $\deg(G) = \alpha < n$ where n is the length of the code. The code C is defined to be the image of an evaluation map ev_D given by

$$\begin{aligned} ev_D : & L(G) \to \mathbb{F}_q^n \\ & f \mapsto (f(P_1), \ldots, (P_n)) \end{aligned} \tag{9.1}$$

The parameters of the code satisfy $k \geq \alpha - g + 1$ and $d \geq n - \alpha$. Recall that the designed minimum distance is $d^* = n - \alpha$.

By the following definition what is called an (e, b)-decodable code can allow an algorithm with a list of at most b elements.

Definition 12. A linear block code C of length n over $\mathbb{F}_q$ is called (e, b)-*decodable* if every Hamming sphere of radius e in $\mathbb{F}_q^n$ contains at most b code words.

The list decoding algorithm finds all these code words. In the next theorem the existence of the list decoding algorithm is stated.

Theorem 10. *Let $C = C_L(D, G)$ be an algebraic geometry $[n, k, d]$-code where D and G are divisors of a curve χ of genus g over $\mathbb{F}_q$. Then, for any positive integer b, C is $(n - \beta - 1, b)$-decodable with*

$$\beta = \left\lceil \frac{(n+1)}{(b+1)} + \frac{b\alpha}{2} + g - 1 \right\rceil \tag{9.2}$$

and

$$\alpha = k + g + 1 \tag{9.3}$$

Before we prove the above theorem, we state the algorithm which is known as Sudan-Guruswami list decoding algorithm for algebraic geometry codes. As we will see later, the output from the algorithm is a list of at most b code words which is within a distance e from a received word a.

Algorithm 5 (List Decoding Algorithm)**.** Let divisors G and F and a vector $a = (a_1, \ldots, a_n)$ be given.

1. Find a non-zero polynomial

$$H(X) = u_b X T^b + \ldots + u_1 X + u_0 \in K[X] \quad \text{where} \quad u_j \in L(F + (b - j)G) \tag{9.4}$$

such that $H(P_i, a_j) = \sum_{j=0}^{b} u_j(P_i) a_i^j$ is zero for $i = 0, \ldots, n$.

2. Find all roots ρ of $H(X)$ in K and calculate

$$x_\rho = (\rho(P_1), \ldots, \rho(P_n))$$

If x_ρ is not defined or if x_ρ differs from a in more than $n - \beta - 1$ coordinates, then this x_ρ is not accepted.

Proof. (Theorem 10) Let $C = C_L(D, G)$ be an algebraic geometry code defined by the curve χ and the evaluation map ev_D as in (9.1).

Let $\deg(G) = \alpha < k + g - 1$ and $D = P_1 + \ldots + P_n$. Define a divisor F on χ so that

$$\deg(F) = \beta - b\alpha = \lceil (n+1)/(b+1) - b\alpha/2 + g - 1 \rceil$$

and the support of F is disjoint from D. Choose an element h in $L(G)$ so that $x = (h(P_1), \ldots, h(P_n))$ is the image of h under the evaluation map ev_D. Let a vector $a = (a_1, \ldots, a_n)$ in $\mathbb{F}_q^n$ be given so that $d(x, a) \geq \beta + 1$, i. e. the vectors a and x agree in at least $\beta + 1$ coordinates.

To complete the proof we look at Algorithm 5 and prove that the output in fact is a list of at most b code words including x. For the algorithm to exist, there must exist a polynomial as in (9.4) satisfying all conditions of step 1. Assume that a basis function is given for each $L(F + (b-j)G)$ where $j = 0, \ldots, b$. The coefficients of u_j with respect to the bases of $L(F + (b-j)G)$ are the unknowns. Hence $H(P_i, a_i)$ is a system of linear equations with

$$\sum_j \dim(F + (b-j)G) = \sum_j \dim(F + jG)$$

unknowns. From Riemann's theorem it follows that

$$\begin{aligned} \dim(F + jG) &\geq \deg(F + jG) - g + 1 = deg(F) + j\alpha - g + 1 = \beta - b\alpha + j\alpha - g + 1 \\ &= \lceil (n+1)/(b+1) - b\alpha/2 + g - 1 \rceil + j\alpha - g + 1 \\ &= \lceil (n+1)/(b+1) - b\alpha/2 \rceil + j\alpha \end{aligned}$$

which implies that

$$\begin{aligned} \sum_{j=0}^{b} \dim(F + jG) &\geq \sum_{j=0}^{b} (\lceil (n+1)/(b+1) - b\alpha/2 \rceil + j\alpha) \\ &= (b+1)\,(\lceil (n+1)/(b+1) - b\alpha/2 \rceil) + \sum_{j=1}^{b} j\alpha \\ &= (n+1) - \lceil (b+1)b\alpha/2 \rceil + (1 + \ldots b)\alpha \end{aligned}$$

Since $(b+1)b/2$ is less than or equal to $(1+\ldots+b)$ for all b we obtain

$$\sum_{j=0}^{b} \dim(F+jG) > n$$

This implies that there are more unknowns than equations, which shows that the set of linear equations has a non-zero solution. Hence there exist a polynomial of the form $H(X) = u_b X^b + \ldots + u_1 X + u_0$. Since the degree of $H(X)$ is b, there exists at most b non-zero roots of the polynomial.

For all roots ρ of $H(X)$, x_ρ is equal to $(\rho(P_1), \ldots, \rho(P_n))$. We want to prove that x is among the set of x_ρ's. Define a set J which contains all values of j which satisfy the condition that $a_j = h(P_j)$. By definition the vector x is $(h(P_1), \ldots, h(P_n))$ and the vector a was constructed under the condition that it agrees with x it at least $\beta+1$ coordinates, thus the size of J is at least $\beta+1$. Hence, $H(h) = u_b h^b + \ldots u_1 h + u_0$ is in $L(F+bG-\sum_{j\in J} P_j)$. But then

$$\begin{aligned}\deg(F+bG-\sum_{j\in J} P_j) &= \deg(F) + b\alpha - \mid J \mid = (\beta - b\alpha) + b\alpha - \mid J \mid \\ &= \beta - \mid J \mid \leq \beta - (\beta+1) = -1 < 0\end{aligned}$$

Thus, $H(h)$ is equal to zero which implies that h is a root of the polynomial H over K which shows that x is among the x_ρ's.

Hence we have proven that for a received word a the Sudan-Guruswami list decoding algorithm returns a list of at most b code words. □

The list decoding algorithm depends on an algorithm for finding the roots of an univariate polynomial over the algebraic function field. This paper will not give an example of such algorithms. For information on how this can be solved see [27].

Remark. When the integer b is equal to 1 the list decoding algorithm can correct $\lfloor (d^* - 1)/2 - g \rfloor$ with complexity $\mathcal{O}(n^3)$. In this case the algorithm falls short of the designed error-correction bound by g. This is quite similar to the result of the basic algorithm. In fact, for $b = 1$ (when the algorithm returns a list of length one) the Sudan-Guruswami algorithm is essentially equal to the basic algorithm. It is possible to modify the algorithm to correct $\lfloor (d^* - 1 - g)/2 \rfloor$ so that the error correction bound agrees with the bound for the basic algorithm.

In general (for other values of b) algebraic geometric codes are (e, b)-decodable for small b and large e. In other words, the error bound for the list decoding algorithm is larger than the designed error bound even when the returned list is quite small.

References

[1] William Fulton, *Algebraic Curves, An Introduction To Algebraic Geometry*, Addison Wesley Publishing Company, 1st Edition, 1969.

[2] David M. Goldschmidt, *Algebraic Functions and Projective Curves*, Springer-Verlag New York, 1st Edition, 2002.

[3] David Cox, John Little, Donald O'Shea, *Using Algebraic Geometry*, Springer-Verlag New York, 1st Edition, 1998.

[4] W. Cary Huffman, Vera Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, 2003.

[5] Robert H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, WILEY, 2002.

[6] J. H. van Lint, *Introduction to Coding Theory*, Springer - Verlag Berlin Heidelberg, 3rd Edition, 1999.

[7] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer - Verlag Berlin Heidelberg, 1st Edition, 2009.

[8] W. Keith Nicholson, *Introduction To Abstract Algebra*, WILEY, 3rd Edition, 2007.

[9] Zhuo Jia Dai, *Algebraic Geometric Coding Theory*, http://upload.wikimedia.org/wikipedia/commons/7/71/Algebraic_Geometric_Coding_Theory.pdf, 2006.

[10] Ernst Kunz, *Introduction To Plane Algebraic Curves*, Birkhäuser Boston, 1991.

[11] Daniel Perrin, *Algebraic Geometry: An Introduction*, Springer - Verlag London, 2008.

[12] I. R. Shafarevich, *Basic Algebraic Geometry*, Springer - Verlag Berlin Heidelberg New York, 1974.

[13] Paulo Ribenboim, *The Riemann-Roch Theorem for Algebraic Curves*, Kingston, Ont. : Queen's University, 2nd Edition, 1965.

[14] Peter Sweeney, *Error Control Coding: From Theory to Practice*, John Wiley & Sons, Ltd., Baffins Lane, England, 2002.

[15] John Baylis, *Error-Correcting Codes: A Mathematical Inroduction* Chapman & Hall, London, 1st Edition, 1998.

[16] Jørn Justesen and Tom Høholdt, *A Course In Error-Correction Codes*, European Mathematical Society, Switzerland, 2004.

[17] Ralf Kötter, *On Algebraic Decoding of Algebraic-Geometry and Cyclic Codes*, Department of Electrical Engeneering, Linköping University, Sweeden, 1996.

[18] Richard A. Mollin, *An Introduction to Cryptography*, Chapman & Hall/CRC, 2nd Edition, 2007.

[19] Tom Høholt and Ruud Pellikaan, *On the Decoding of Algebraic-Geometric Codes* IEEE Transaction of Information Theory, Volume 41, 1995.

[20] Iwan M. Duursma, *Algebraic Decoding Using Special Divisors*, IEEE Transaction of Information Theory, Volume 32, 1993.

[21] Alexei N. Skorobogatov and Sergei G. Vladut, *On the Decoding of Algebraic-Geometric Codes*, IEEE Transaction of Information Theory, Volume 36, 1990.

[22] Kyoki Imamura and Wataru Yoshida, *A Simple Derivation of the Berlekamp-Massey Algorithm*, IEEE Transaction of Information Theory, Volume 33, 1987.

[23] Y. Sudiyama, M. Kasahara, S. Hirasawa and T. Namekawa, *A Method for Solving Key Equation for Decoding Goppa Codes*, IEEE Transaction of Information Theory, Volume 27, 1975.

[24] Doug Ierardi and Ming-Deh Huang, *Efficient Algorithm for the Riemann-Roch Problem and for Addition in the Jacobian of a curve*, Journal of Symbolic Computaion, 18:519-539, 1994.

[25] Joseph H. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag New York, 2nd Edition, 2009.

[26] Henri Cohen and Gerhard Frey, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Chapman & Hall/CRC, 2006.

[27] M. Amin Shokrollahi and H. Wasserman, *List Decoding of Algebraic-Geometric Codes*, IEEE Transaction of Information Theory, Volume 45, No. 2, 1999.

[28] Ralf KŽtter, *A Fats Parallel Implementation of a Berlekamp-Massey Algorithm for Algebraic-Geometric Codes*, IEEE Transaction of Information Theory, Volume 44, No. 4, 1998.