On Algebraic Geometric Codes and Some Related Codes

Kenza Guenda

Thesis presented in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE AT

 STELLENBOSCH UNIVERSITY

SUPERVISED BY: Prof. Barry W GREEN

December, 2006
Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own work and has not previously, in its entirety or in part, been submitted at any other university for a degree.

Signature:..........................
Date:..............................
Abstract

The main topic of this thesis is the construction of the algebraic geometric codes (Goppa codes), and their decoding by the list-decoding, which allows one to correct beyond half of the minimum distance. We also consider the list-decoding of the Reed–Solomon codes as they are subclass of the Goppa codes, and the determination of the parameters of the non primitive $BCH$ codes.

AMS Subject Classification: 4B05, 94B15, 94B35, 94B27, 11T71, 94B65, B70.

Keywords: Linear codes, cyclic codes, $BCH$ codes, Reed–Solomon codes, list-decoding, Algebraic Geometric codes, decoding, bound on codes, error probability.
Opsomming

Die hoofonderwerp van hierdie tesis is die konstruksie van die algebraïese-meetkundige kodes (Goppa Kodes) en die ontsyfering daarvan met behulp van die "list-decoding." Hierdie metode stel ons in staat om meer foute as die helfte van die minimum afstand tussen kodeerwoorde te korrigeer. Ons ondersoek ook die "list-decoding" van die Reed-Solomon kodes, ’n onderfamilie van die Goppa kodes en bepaal ook die parameters van nie-primitiewe BCH kodes.”
Acknowledgement

I am greatly indebted to the African Institute of Mathematical Sciences and to the University of Stellenbosch for the grant that supported my study.

I am grateful to my supervisor Professor Barry W Green, whom I thank for the choice of the subject and for his guidance in my research during the last 18 months, and from who I learnt a lot.

Special thanks to the Professors Lemmeounar Noui, Schehrazad Selmane and to all my friends and colleagues for their advice and their support.

I was so fortunate to meet the Professors Helmut Prodinger and Hans-Georg Rück and to have the email contact with Karim Belabas, and I am grateful for their scientific generosity. Thank you Georg, Karim, and Helmut.

Many thanks to the students with whom I was sharing the office and the team spirit, especially to Ilhem Benzaoui whom I thank for being my friend and for mathematical discussions, to Yves Semegni for the initiation to Mathematica, to Richard Akinola and Deter DeWet for the English help.

Thanks to Linda Uys from the International Office for her help.
A warm thanks to my family for their support, encouragement and the belief in my success.

Kenza Guenda
# Contents

Abstract i  
Opsomming ii  
Acknowledgement iii  
Abbreviation and Notation ix  

1 Introduction 1  

2 Linear Codes 4  

2.1 Preliminaries and Definitions 4  
2.2 The Decoding Rules 6  
  2.2.1 NP-Hardness of the $MLD$ 8  
  2.2.2 The Bounded Distance Decoding 8  
2.3 Parameters Bound 9  
2.4 Asymptotic Bounds and Good Linear Codes 11  

3 Reed–Solomon and $BCH$ Codes 14  

3.1 Reed–Solomon Codes 14
3.2 Finite fields ................................................. 15
  3.2.1 The Roots of Unity .................................. 15
  3.2.2 Factoring $x^n - 1$ ................................. 16

3.3 Cyclic Codes ............................................... 21

3.4 $BCH$ Codes ............................................... 22
  3.4.1 Dimension of $BCH$ Codes ......................... 23
  3.4.2 The Minimum Distance .............................. 24
  3.4.3 Can our $BCH$ Codes be Self-Orthogonal .......... 30

3.5 Cyclic Reed-Solomon Codes .............................. 31

4 The List-Decoding Algorithm for Reed–Solomon Codes 33
  4.1 Sudan’s Algorithm for Reed Solomon codes .......... 35
  4.2 Guruswami–Sudan Algorithm .......................... 41
    4.2.1 The Asymptotic Bound on the Radius .............. 46
    4.2.2 The Run time of the Algorithm ..................... 48

5 The Factorization Step in the List-Decoding Algorithm 49
  5.1 The Factorization of the Interpolation Polynomial .. 50
  5.2 A Solution to the Factorization Step in the List-Decoding Algorithm ........................................ 51
  5.3 The Cost of the Algorithm .............................. 55

6 Goppa Codes: Construction and Properties 58
  6.1 Some Notions from Algebraic Geometry ............ 58
    6.1.1 Characterisation of the Singular Points of an Algebraic Curve ........................................ 63
6.1.2 Divisors on Algebraic Curves  
6.1.3 Linear Systems  
6.2 Construction of the Goppa Codes  
6.3 The Goppa Code Family is a Good Family  
6.3.1 Number of Rational Points  
6.3.2 Asymptotic Bounds  

7 List Decoding Algorithm for Goppa Codes  
7.1 Description of the Algorithm  
7.1.1 Factorization Step in the List-Decoding Algorithm for AG Codes  
7.1.2 Explicit List-Decoding Algorithm for AG Codes  

8 Conclusion  
8.1 Contribution of this Work  
8.1.1 The Parameters of the non-Primitive BCH Codes  
8.1.2 Presentation of the List-Decoding  
8.1.3 The Solution for the Factorization Step in the list-Decoding Algorithm  
8.2 Problems for Further Research  

vii
## Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_q$</td>
<td>Finite field with $q$ elements</td>
<td>4</td>
</tr>
<tr>
<td>$R$</td>
<td>Rate of a linear code</td>
<td>4</td>
</tr>
<tr>
<td>$d$</td>
<td>Minimum Hamming-distance of a code $C$</td>
<td>5</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Minimum weight of a code $C$</td>
<td>5</td>
</tr>
<tr>
<td>$C^\perp$</td>
<td>Dual code of a code $C$</td>
<td>5</td>
</tr>
<tr>
<td>$x^T$</td>
<td>Vector transpose of $x$</td>
<td>6</td>
</tr>
<tr>
<td>$BSC$</td>
<td>Binary symmetric channel</td>
<td>6</td>
</tr>
<tr>
<td>$MLD$</td>
<td>Maximum likelihood decoding</td>
<td>7</td>
</tr>
<tr>
<td>$MDS$</td>
<td>Maximum-distance-separable</td>
<td>8</td>
</tr>
<tr>
<td>$\alpha_q(\delta)$</td>
<td>Asymptotic rate</td>
<td>11</td>
</tr>
<tr>
<td>$R_{GV}$</td>
<td>Asymptotic Gilbert–Varshamov bound</td>
<td>11</td>
</tr>
<tr>
<td>$H_q(x)$</td>
<td>$q$-ary entropy function</td>
<td>12</td>
</tr>
<tr>
<td>$RS$</td>
<td>Reed–Solomon codes</td>
<td>13</td>
</tr>
<tr>
<td>$L_k$</td>
<td>Vector space of the $\mathbb{F}_q$ polynomials of degree less than $k$</td>
<td>14</td>
</tr>
<tr>
<td>$Cl(j)$</td>
<td>Cyclotomic class</td>
<td>17</td>
</tr>
<tr>
<td>$T$</td>
<td>Defining set of a cyclic code</td>
<td>22</td>
</tr>
<tr>
<td>$B_\tau(r)$</td>
<td>Hamming ball</td>
<td>35</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Number of errors corrected</td>
<td>36</td>
</tr>
<tr>
<td>$[n]$</td>
<td>The set ${1, \ldots, n}$</td>
<td>36</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>$\mathbb{A}^n$</td>
<td>Affine space of dimension $n$ over $\mathbb{F}_q$</td>
<td></td>
</tr>
<tr>
<td>$\mathbb{P}^n$</td>
<td>Projective space of dimension $n$</td>
<td></td>
</tr>
<tr>
<td>$I(V)$</td>
<td>Ideal of an Algebraic set $V$</td>
<td></td>
</tr>
<tr>
<td>$\text{Gal}(\overline{\mathbb{F}_q}/\mathbb{F}_q)$</td>
<td>Galois group of the extension $\overline{\mathbb{F}_q}/\mathbb{F}_q$</td>
<td></td>
</tr>
<tr>
<td>$V(\mathbb{F}_q)$</td>
<td>Subset of $\mathbb{F}_q$-rational points of a variety $V$</td>
<td></td>
</tr>
<tr>
<td>$\overline{\mathbb{F}_q}[V]$</td>
<td>Homogeneous coordinate ring of $V$</td>
<td></td>
</tr>
<tr>
<td>$\overline{\mathbb{F}_q}(V)$</td>
<td>Function field of $V$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{C}$</td>
<td>Projective curve</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{O}_P(V)$</td>
<td>Local ring of the point $P$</td>
<td></td>
</tr>
<tr>
<td>$M_P(V)$</td>
<td>Maximal ideal of the local ring of the point $P$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{J}_{V,P}$</td>
<td>Jacobian matrix at the point $P$ of the variety $V$</td>
<td></td>
</tr>
<tr>
<td>$v_P$</td>
<td>Valuation of the local ring $\mathcal{O}_P$</td>
<td></td>
</tr>
<tr>
<td>$F_P$</td>
<td>Residue class field</td>
<td></td>
</tr>
<tr>
<td>$D$</td>
<td>Divisor</td>
<td></td>
</tr>
<tr>
<td>$\text{supp } D$</td>
<td>Support of the divisor $D$</td>
<td></td>
</tr>
<tr>
<td>$\text{deg } D$</td>
<td>Degree of the divisor $D$</td>
<td></td>
</tr>
<tr>
<td>$(f)$</td>
<td>Principal divisor of $f$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{L}(G)$</td>
<td>Vector space associated with divisor $D$</td>
<td></td>
</tr>
<tr>
<td>$l(G)$</td>
<td>Dimension of the vector space $\mathcal{L}(D)$</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>Genus of a curve $\mathcal{C}$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{C}_G(D,G)$</td>
<td>Geometric Goppa code</td>
<td></td>
</tr>
<tr>
<td>$Z_{\mathcal{C}}(t)$</td>
<td>Zeta function associated with the curve $\mathcal{C}$</td>
<td></td>
</tr>
<tr>
<td>$N_q$</td>
<td>Number of $\mathbb{F}_q$-rational point over $\mathcal{C}$</td>
<td></td>
</tr>
<tr>
<td>$Z^{(s)}(t)$</td>
<td>s- Zeta function associated to the curve $\mathcal{C}$</td>
<td></td>
</tr>
<tr>
<td>$K$</td>
<td>Function field $\mathbb{F}_q(\mathcal{C})$ of the curve $\mathcal{C}$</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In the beginning is the truth .
Ibn El Kaem .

Error-correcting codes (or simply, codes) are objects designed to cope with the problem of information transmission on a noisy channel. This theory was created by Shannon in 1948 [34], when he proved the existence of good families of codes. Even though young, it was rapidly developed by taking benefit from techniques developed in a wide variety of topics, such combinatorics, probability, algebra, geometry, engineering, and computer science. In turn error-correcting codes has diverse applications in a variety of areas. In the 1970’s D. Goppa discovered an important link between coding theory and algebraic geometry. This discovery led to the construction of powerful new error-correcting codes using techniques from algebraic geometry in particular, the theory of curves (or equivalently function fields) over finite fields. These codes are called geometric Goppa codes or algebraic geometric codes or simply AG codes. In 1982 Tsfasman, Vladut and Zink [41], constructed the first class of Goppa codes which are above the Gilbert–Varshamov bound (the bound which measures how good a family of codes is). The Goppa codes are also used in cryptography and they were chosen by McEliece for his cryp-
tosystem, which is still considered secure. A very special case of Goppa codes are the Reed–Solomon codes, which are prevalent in practical applications. They are used to store information, music and video on compact discs and digital video discs.

In the first part of this thesis, we introduce linear codes and their parameters, since Goppa codes are also linear codes. The decoding rules and the asymptotic bounds of decoding are treated with the aim to understand the importance of the Goppa codes.

In the third chapter we treat the Reed-Solomon codes which can be considered as Goppa codes over the projective line. Using the properties of the cyclotomic classes, in this chapter the parameters of the non primitive BCH codes, or “an interesting non primitive BCH codes” are determined. The Reed–Solomon codes can be viewed also as a particular case of the BCH codes. In this chapter all the results and proofs which are given without references are original. We used the software Mathematica to compare the bound given for the minimum distance with the Griesmer bound. At the end of the chapter we give the link between the BCH codes and the Reed–Solomon codes.

The list-decoding algorithm for the Reed–Solomon codes is given in the fourth chapter. This algorithm allows us to decode beyond half of the minimum distance, which is done by conventional algorithms. The list-decoding algorithm contains two essentials steps, the interpolation step and the factorization step.

When any algorithm is designed the running time is always the first characteristic of the algorithm. In chapter 5 we give a new way of factorizing the interpolation polynomial given by the interpolation step in the list-decoding algorithm. We compare the running time of the algorithm proposed to the running time of some other algorithms.

In the chapter 6 we introduce some notions of algebraic geometry, affine spaces, projective varieties and divisors on an algebraic curve. These notions
allow us to construct the Goppa codes or algebraic geometric codes, by $\mathcal{L}$-construction. We ignore the $\omega$ construction which gives the dual of the codes constructed by the $\mathcal{L}$-construction. In this chapter, we also give the parameters of these codes by using the Riemann–Roch theorem. At the end of the chapter we show how the Goppa family is a good family, by discussing the more recent results given in [42], [37] and [49].

Finally in the last chapter we give the list-decoding algorithm for the Goppa codes.
Chapter 2

Linear Codes

We must know, we will know .

D.Hilbert

2.1 Preliminaries and Definitions

Let $\mathbb{F}_q$ be a finite field with $q$ elements. We consider the $n$-dimensional space $\mathbb{F}_q^n$ whose elements are $n$-tuples $x = (x_1, \ldots, x_n)$ with $x_1, \ldots, x_n \in \mathbb{F}_q$.

Definition 1 A code $C$ of length $n$ is any non-empty subset of $\mathbb{F}_q^n$. The elements of $C$ are called \textbf{codewords}. The code is called \textbf{linear} if it is a $\mathbb{F}_q$-subspace of $\mathbb{F}_q^n$, and then the dimension $k$ of $C$ is called the \textbf{dimension} (or \textbf{rank}) of the code.

The \textbf{transmission rate} or simply rate of a linear code is the quantity $R = k/n$. The matrix $G$ that has as its rows the $\mathbb{F}_q^n$ basis vectors of $C$, is called a \textbf{generator matrix} of the code $C$.

For $x, y \in \mathbb{F}_q^n$, a metric $d$ called the \textbf{Hamming distance} is defined between $x$ and $y$ by

$$d(x, y) = |\{i \mid x_i \neq y_i\}|.$$
where $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$.

The **weight** of $x$ is defined by

$$w(x) = d(x, 0) = |\{i \mid x_i \neq 0\}|,$$

where $x = (x_1, \ldots, x_n)$. The **minimum distance** $d = d(C)$ of a code $C \subset \mathbb{F}_q^n$ is given by

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

and the **minimum weight** of $C$ is

$$\omega(C) = \min \{\omega(x) \mid x \in C, x \neq 0\}.$$

For a linear code we have $d(C) = w(C)$.

The rational number $\delta(C) = d(C)/n$ is called the **relative distance** of $C$.

A linear code $C$ of length $n$, dimension $k$ and minimum distance $d$ over $\mathbb{F}_q$ is called an $[n, k, d]_q$-code.

**Definition 2** Let $C \subset \mathbb{F}_q^n$ be a linear code of dimension $k$. The **dual code** of $C$ is the code $C^\perp$ defined by

$$C^\perp = \{x \in \mathbb{F}_q^n \mid \langle x, y \rangle = 0, \forall y \in C\},$$

where for $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$, $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ is the usual bilinear form on $\mathbb{F}_q^n \times \mathbb{F}_q^n$.

The dual code $C^\perp$, is a linear code of length $n$ and dimension $k^\perp = n - k$. Its generator matrix $H$, called the **parity check matrix** of $C$, verifies

$$C = \{x \in \mathbb{F}_q^n \mid Hx^T = 0\}. \quad (2.1)$$
2.2 The Decoding Rules

The model behind these concepts is the following. A message consisting of elements from $\mathbb{F}_q$ is to be transmitted via some medium (called the channel). The channels considered are usually communication channels or storage devices. Since an $[n, k]$ code $C$ is $\mathbb{F}_q$-linear of dimension $k$, it contains $q^k$ codewords and can be used to send $q^k$ distinct messages. Before the message is sent it is first divided into message words of length equal to the rank $k$ of $C$ and then each message word is transformed into a codeword by multiplying it by the generator matrix of $C$. This operation is called encoding the message word and costs $\mathcal{O}(nk)$ field operations. After encoding, the transmission is $R = k/n$ times slower, which justifies the term information rate. Because of the channel noise the codeword $x$ which is sent, is received as the vector $y$ and the decoder must decide what the error vector $e = y - x$ is.

**Definition 3** A communication channel is said to be **memoryless** if the outcome of any one transmission is independent of the outcome of the previous transmissions; i.e., if $x = x_1 \ldots x_n$ and $y = y_1 \ldots y_n$ are vectors of length $n$ over $\mathbb{F}_q$, then

$$P(y \text{ received } | x \text{ sent}) = \prod_{i=1}^{n} P(y_i \text{ received } | x_i \text{ sent})$$

**Definition 4** A $q$-ary symmetric channel (qSC) is a memoryless channel, over $\mathbb{F}_q$ such that

1. each outcome transmission has the same probability $p < 1/q$ (called the crossover probability) of being received in error.

2. if a symbol is received in error, then each of the $q - 1$ possible errors is equally likely.
3. the channel probability is given by

\[ P(y \text{ received} \mid x \text{ sent}) = p \text{ for } y \neq x; \quad P(x \text{ received} \mid x \text{ sent}) = 1 - (q-1)p \]

**Example 5** The binary symmetric channel (BSC) is a memoryless channel, which has channel alphabet \{0, 1\}, and channel probabilities,

\[ P(1 \text{ received} \mid 0 \text{ sent}) = P(0 \text{ received} \mid 1 \text{ sent}) = p \]

\[ P(1 \text{ received} \mid 1 \text{ sent}) = P(0 \text{ received} \mid 0 \text{ sent}) = 1 - p \]

The **Maximum likelihood decoding** (MLD) rule is based on the idea that \( c_y \) is the most likely codeword sent if

\[ P(y \text{ received} \mid c_y \text{ sent}) = \max_{c \in C} P(y \text{ received} \mid c \text{ sent}). \]

For a \( qSC \) if \( c_y \) and \( y \) differ in \( i \) places then

\[ P(y \text{ received} \mid c_y \text{ sent}) = p \left[ 1 - (q-1)p \right]^{n-i}. \]

Since \( p < 1/q \) this quantity is maximized if \( i \) is minimized. For a linear code we have defined the Hamming distance \( d(x, y) \) between two vectors \( x, y \) of the code \( C \) by the number of places in which \( x \) and \( y \) differ, and then if we use a \( q \)-ary symmetric channel decoding by **minimum-distance** or **nearest-neighbor** decoding, which means decoding a received vector \( y \) into a codeword \( x \) that is a minimum distance from \( y \) is the same as using the MLD rule. Often the codeword found is not unique.
2.2.1 NP-Hardness of the MLD

It was shown by Berlekamp et al [5] that the MLD for a general binary linear code is an NP-hard problem, meaning that an efficient algorithm for solving this problem in a time complexity that can be expressed as a polynomial in the code length $n$ is not likely to exist. Nevertheless it was unknown whether the MLD remains hard for specific families of codes. Recently Guruswami and Vardy have shown in [16] that the MLD is NP-hard for the Reed–Solomon codes.

A potential way to attempt to circumvent the NP-hardness of the MLD is to correct only a limited number of errors. In order to do so, the MLD is parametrized to the reasonable decoding problem given in the following paragraph.

2.2.2 The Bounded Distance Decoding

The bounded distance decoding (BDD) is as follows:

Given a specific linear code $C$, a received vector $y$ in $\mathbb{F}_q^n$, and a positive integer $\tau$ find any codeword $x$, such that

$$d(x, y) \leq \tau,$$

if such codeword exists.

The following lemma gives an upper bound on the number of errors corrected using the BDD rule and gives the strategy called decoding up to the minimum distance.

Lemma 6 A linear code $C$ with minimum distance $d$ can correct up to $\tau$ errors in any codeword if $d \geq 2\tau + 1$.

Proof. Suppose $d \geq 2\tau + 1$. Suppose a codeword $x$ is transmitted and the vector $y$ is received for which $\tau$ or fewer errors have occurred, so that
\(d(x, y) \leq \tau\). If \(x'\) is any codeword other than \(x\), then \(d(x', y) \geq \tau + 1\). For otherwise \(d(x', y) \leq \tau\) implies, by the triangle inequality, that \(d(x, x') \leq d(x, y) + d(x', y) \leq 2\tau\), contradicting \(d \geq 2\tau + 1\). So \(x\) is the nearest codeword to \(y\) and nearest neighbour decoding corrects the errors. \(\blacksquare\)

This means if we take \(\tau \leq \lfloor (d - 1)/2 \rfloor\) then for any \(x \in \mathbb{F}_q^n\), there is at most one codeword \(c\) in the Hamming ball \(B_\tau(x)\), where the Hamming ball of center \(x\) and radius \(\tau\) is defined as

\[
B_\tau(x) = \{ c \in \mathbb{F}_q^n \mid d(x, c) \leq \tau \},
\]

implying that the Hamming ball around the codewords are all disjoint if their radius is \(\tau \leq \lfloor (d - 1)/2 \rfloor\). It follows that when more than \(\lfloor (d - 1)/2 \rfloor\) errors occur, the decoder either declares an error or gives as output a codeword different from the transmitted one. This decoding strategy will be referred to as the conventional approach.

The decoding beyond the half of the minimum distance will be the subject of chapter 4.

### 2.3 Parameters Bound

There are some restrictions on the parameters \(n, k, d, q\) of a linear code. The simplest one is the **Singleton bound**

\[k + d \leq n + 1.\]

If the Singleton bound is attained, the code is called **Maximum-Distance-Separable (MDS)**. Since from lemma 6, the minimum distance is related to the correctability of the code, the codes for which this bound is attained are optimum in this sense.
The Singleton bound is the linear case for the following bound.

**Theorem 7** *(Griesmer Bound)* \([23]\) **p547** Let \(C\) be an \([n,k,d]\) code over \(\mathbb{F}_q\). Then

\[
n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil
\]

Since \([d/q^0] = d\) and \([d/q^i] \geq 1\) for \(i \in [k-1]\), the Griesmer bound implies the Singleton bound.

Another important bound which relates the parameters and solves one of the most important problems of coding theory: For given parameters \(n, k, d\) is there an \([n,k,d]\) code over \(\mathbb{F}_q\)?

The following theorem often called the **sphere-packing** or **Hamming-bound** gives the answer.

**Theorem 8** \(([18]\)Theorem 1.12.1) For all integers \(n, k, d\), an \([n,k,d]\) code over \(\mathbb{F}_q\) exist only if

\[
q^k \sum_{i=0}^{\left\lfloor \frac{d-1}{2} \right\rfloor} \binom{n}{i} (q-1)^i \leq q^n
\]

There is another important bound which is given in the following theorem.

**Theorem 9** *Gilbert-Varshamov* \([30]\)Th.4.5.5

Let \(n,k,d\) be integers satisfying \(2 \leq d \leq n\) and \(1 \leq k \leq n\). If

\[
\sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i < q^{n-k},
\]

then there exists an \([n,k]\)–linear code over \(\mathbb{F}_q\) with minimum distance at least \(d\).
2.4 Asymptotic Bounds and Good Linear Codes

By the Singleton-bound we have \( k \leq n - d + 1 \) and by dividing by \( n \), we get

\[
R = \frac{k}{n} \leq 1 - \delta + \frac{1}{n},
\]

(2.2)

where \( \delta = \frac{d}{n} \). If we keep \( \delta \) fixed and let \( n \) go to infinity then we see that,

\[
\alpha_q(\delta) \leq 1 - \delta, \text{where } \alpha_q(\delta) = \limsup R.
\]

\( \alpha_q(\delta) \) is called the asymptotic rate corresponding to \( 0 \leq \delta \leq 1 \).

Manin in [24] proved that \( \alpha_q(\delta) = 0 \) for \( 1 - q^{-1} \leq \delta \leq 1 \) and \( \alpha_q(\delta) \) is decreasing in the interval \( 0 \leq \delta \leq 1 - q^{-1} \). In general the asymptotic rate \( \alpha_q(\delta) \), is defined as \( \limsup_{n \to \infty} \log_q A_q(n, d)/n \), where

\[
A_q(n, d) = \max\{q^k \mid \text{there exist an } [n, k, d] \text{ code over } \mathbb{F}_q\}. 
\]

The exact value of \( \alpha_q(\delta) \) is unknown and hence we want to find upper and lower bounds. An upper bound on \( \alpha(\delta) \) would indicate that all families with relative distances approaching \( \delta \) have rates in the limit at most this upper bound. A lower bound on \( \alpha_q(\delta) \) indicates that there exists a family of codes of lengths approaching infinity and relative distances approaching \( \delta \) whose rate are at least this bound. The inequality (2.2) gives us that the Singleton-bound is an upper bound; there are more important upper bounds cited in [43], [18], [48] and [42]. The most important lower bound is the bound given by the following theorem:

Theorem 10 Asymptotic Gilbert-Varshamov bound

If \( 0 < \delta \leq 1 - \delta^{-1} \) where \( q \geq 2 \), then

\[
\alpha_q(\delta) \geq 1 - H_q(\delta) = R_{GV}.
\]
where $H_q(x)$ is the $q$-ary entropy function which is defined as follows:

for $0 \leq x \leq 1$

$$H_q(0) = 0,$$

$$H_q(x) = x \log_q(q - 1) - x \log_q(x) - (1 - x) \log_q(1 - x)$$

(2.3)

The $q$-ary entropy function arises here from the fact that the

$$\limsup_{n \to \infty} \frac{1}{n} \log_q\left(\sum_{i=0}^{d} \binom{n}{i} (q-1)^i\right) = H_q(\delta).$$

The proof of the asymptotic Gilbert-Varshamov bound is very simple and based essentially on the result of the Theorem 9 and an approximation using Stirling formula. This bound which was discovered in 1952 by Gilbert and Varshamov\(^1\), guarantees (theoretically) the existence of families $[n_i, k_i, d_i]$ of codes with increasing length, such that for a given $\epsilon > 0$ there exists $j_\epsilon \in \mathbb{N}^*$ such that

$$\frac{d_j}{n_j} \geq \delta, \quad \frac{k_j}{n_j} \geq 1 - H_q(\delta) - \epsilon$$

for all $j > j_\epsilon$, in other words the sequence $\delta_j$ approaches $\delta$ as the rate approaches $R_{GV} = 1 - H_q(\delta)$. A family with this property is said to meet the Gilbert-Varshamov-bound. A family is said **asymptotically good**\(^2\) if it satisfies the following weaker property:

There exist positive real numbers $R$ and $\delta$ such that $k_i/n_j \geq R$ and $d_j/n_j \geq \delta$ for all $j$. A family which is not asymptotically good is said to be **asymptotically bad**, and in this case either $k_j/n_j$ or $d_j/n_j$ approach 0 when

\(^1\)In fact Gilbert in [13] showed essentially that a random code has this property, Varshamov [46] showed that random linear codes have this property, and Wozencraft [47] constructed a small space of linear codes most of whose members meet the Gilbert-Varshamov bound.

\(^2\)The existence of good families was proved in [34], but the theorem is probabilist not constructive.
$j \rightarrow \infty$. Roughly speaking a good family is a family with large rate (for efficiency) and large minimum distance to correct the maximum number of errors. For more than 30 years it was conjectured that $\alpha_q(\delta)$ would coincide with $R_{GV} = 1 - H_q(\delta)$, because no one gave any family that exceed this bound. In chapter 6 we will see that the conjecture was false by using the Goppa codes.
Chapter 3

Reed–Solomon and $BCH$ Codes

3.1 Reed–Solomon Codes

The Reed–Solomon codes, abbreviated $RS$ codes, are possibly the most commonly used codes in practical applications. In particular they are used to store information, music and video on compact discs (CDs) and digital video discs (DVDs). Further, they have good parameters (they are $MDS$) and have efficient algorithms for decoding$^1$.

The $RS$ code can be constructed as follows.

Let $\alpha_1, \ldots, \alpha_n$ be $n$ distinct elements of $\mathbb{F}_q$. For $k \in \mathbb{N}$ consider the vector space

$$L_k = \{ f(x) \in \mathbb{F}_q[x] \mid \deg f \leq k - 1 \}.$$

The dimension of $L_k$ is $k$. For $k \leq n \leq q$ a non-zero polynomial $f(x) \in L_k$

\footnote{This family is asymptotically bad as it was shown in theorem 3 Ch.10 [23], but can be used to construct a good family by concatenation and some other transformations.}.
cannot vanish at all the \( \alpha_i \). Hence if \( n \geq k \), the evaluation map

\[
ev : \mathcal{L}_k \rightarrow \mathbb{F}_q^n
\]

\[
f \mapsto (f(\alpha_1), \ldots, f(\alpha_n)),
\]

is injective, and its image \( C \) is an \([n,k]\) MDS-code, called a Reed–Solomon code. The fact that it is an MDS-code will be seen in section 6 of this chapter. However, these codes have a drawback, that their length \( n \) is restricted to be less than or equal to \( q \), the size of \( \mathbb{F}_q \). This is not desirable for many applications where codes over small alphabet are required.

Since \( 1, x, \ldots, x^{k-1} \) is a basis of \( \mathcal{L}_k \), a generator matrix of \( C \) is the following:

\[
\begin{pmatrix}
1 & \cdots & 1 \\
\alpha_1 & \cdots & \alpha_n \\
\alpha_1^{k-1} & \cdots & \alpha_n^{k-1}
\end{pmatrix}
\]

In the next part of this chapter we will be interested to define the properties of a class of codes such that the Reed–Solomon codes are a subclass. For that we need to recall some properties of finite fields.

### 3.2 Finite fields

#### 3.2.1 The Roots of Unity

Let \( \mathbb{F}_q \) be the finite field with \( q \) elements, and \( n \) an integer such that \((n, q) = 1\).

The splitting field of \( x^n - 1 \) over \( \mathbb{F}_q \) is denoted by \( \mathbb{F}_{q^*} \). The polynomial \( x^n - 1 \) has no multiple roots in any extension of \( \mathbb{F}_q \), since its derivative \( D[x^n - 1] = nx^{n-1} \), has no common factor with \( x^n - 1 \). Hence \( x^n - 1 \) has \( n \)
roots in its splitting field $\mathbb{F}_{q^s}$, which we call the $n$-th roots of unity over $\mathbb{F}_q$.

The set $E^{(n)}$ of $n$-th roots of unity is a cyclic subgroup of size $n$ of the multiplicative group $\mathbb{F}_{q^s}^*$, a generator of $E^{(n)}$ is called a primitive $n$-th root of unity over $\mathbb{F}_q$ (i.e., a primitive $n$-th root of unity is of order $n$).

**Theorem 11** ([30] p309) If $\mathbb{F}_{q^s}$ is the splitting field of $x^n - 1$ over $\mathbb{F}_q$, then $s$ is the smallest positive integer for which $n$ divides $q^s - 1$, that is, $s$ is the smallest positive integer for which $q^s \equiv 1 \mod n$ i.e., $s$ is the order of $q$ modulo $n$, which is denoted $\text{ord}_n q$.

**Theorem 12** ([30] p310) Let $\beta$ be a primitive element of the splitting field $\mathbb{F}_{q^s}$ of $x^n - 1$. Then the $\phi(n)$ primitive $n$-th root of unity over $\mathbb{F}_q$ are precisely the elements

$$\left\{\beta^k \mid k = \frac{q^s - 1}{n} u, \quad u < n, (u, n) = 1\right\}.$$

3.2.2 Factoring $x^n - 1$

From theorem 12, if we have a primitive element $\beta$ of $\mathbb{F}_{q^s}$, where $s = \text{ord}_n q$, we obtain a primitive $n$-th root of unity

$$\alpha = \beta^{(q^s - 1)/n} \quad (3.2)$$

Hence the roots of $x^n - 1$ are given by

$$1, \alpha, \ldots, \alpha^{n-1}$$

and then

$$x^n - 1 = \prod_{i=0}^{n-1} (x - \alpha^i). \quad (3.3)$$

Since the roots of $x^n - 1$ are distinct, then $x^n - 1$ is just the product of the minimal polynomials of its roots. We recall the definition of minimal polynomial.
Definition 13 Let $\gamma \in \mathbb{F}_{q^s}$. The minimal polynomial of $\gamma$ is the lowest degree monic polynomial denoted $M_\gamma(x) \in \mathbb{F}_q[x]$, such that $M_\gamma(\gamma) = 0$.

Proposition 14 ([23] p106) Let $\gamma \in \mathbb{F}_{q^s}$, and $M_\gamma(x)$ be its minimal polynomial. Then $\deg M_\gamma(x) = d$ if and only if $d$ is the smallest positive integer such that $\gamma^{q^d} = \gamma$.

From the definition of the order of $q$ modulo $l$ and the proposition 14 we can deduce the following corollary.

Corollary 15 If $\gamma \in \mathbb{F}_{q^s}$ such that $l$ is the order of $\gamma$ then

$$\deg M_\gamma(x) = \text{ord}_l q$$

the following proposition gives the form of $M_\gamma(x)$.

Proposition 16 ([30] p310) Let $\alpha$ be an $n$-th primitive root of unity of $\mathbb{F}_{q^s}$ and $\gamma = \alpha^j$. Then the minimal polynomial of $\gamma$ is given by

$$M_\gamma(x) = \prod_{i=0}^{\text{ord}_l q - 1} (x - \alpha^{jq^i}).$$

In this way we can denote $M_\gamma(x)$ simply by $M_j(x)$. To the set $\{\alpha^j, \alpha^{jq}, \ldots, \alpha^{jq^{\text{ord}_l q - 1}}\}$ we can associate the set $Cl(j) = \{j, jq, \ldots, jq^{\text{ord}_l q - 1}\}$mod $n$, called the cyclotomic class of $j$ modulo $n$, and then $r = \text{ord}_l q$ is the smallest positive integer such that $jq^r = j$mod $n$. The integers $j$ and $l$ are related by the fact that $l$ is the order of $\gamma = \alpha^j$ i.e.,

$$l = \frac{n}{(n, j)}.$$
From the proposition 16 and the equation (3.3) we obtain the following factorization of $x^n - 1$.

$$x^n - 1 = \prod_{j \in K} M_j(x)$$

where $K$ is a set of representatives modulo $n$.

**Lemma 17** For all $j \in \mathbb{Z}_n := \{0, \ldots, n - 1\}$, the cardinality of $Cl(j)$ is a divisor of the cardinality of $Cl(1)$, and $\deg M_j$ divides $\deg M_1$.

**Proof.** Let $j \in \mathbb{Z}_n$ such that $\gamma = \alpha^j$ from corollary 15, we have

$$\deg M_j = \operatorname{ord} \ i q$$

where $l = \frac{n}{(n,j)}$ is the order of $\gamma$. Since $l$ divides $n$, then $\operatorname{ord} \ i q$ divides $\operatorname{ord} \ n q$. But $\operatorname{ord} \ n q = |Cl(1)| = \deg M_j$. Hence the result follows. ■

**Definition 18** Let $i \in \mathbb{Z}_n$ and $Cl(i)$ be the cyclotomic class of $i$ modulo $n$ over $\mathbb{F}_q$. Then $Cl(i)$ is called reversible if it satisfies,

$$Cl(i) = Cl(-i).$$

**Lemma 19**

*If $Cl(1)$ is reversible, then $\forall j \in \mathbb{Z}_n, Cl(j)$ is reversible.*

**Proof.** Assume that $Cl(1)$ is reversible, then there exists $k$, $1 \leq k \leq \operatorname{ord} \ n q$, such that

$$q^k \equiv -1 \mod n$$

Let $j \in \mathbb{Z}_n$ then

$$jq^k = -j \mod n$$
is $k$ such that $jq^k \in Cl(j)$ i.e., is there $t \leq \text{ord}_{jq}$, where $l = \frac{n}{(n,j)}$?

We have $k \equiv i \mod \text{ord}_{jq}$, then

if $i > 0$, it suffices to take $t = i$,

if $i < 0$ it suffices to take $t = i + \text{ord}_{jq}$.

Lemma 20  For $n = q^m + 1$, with $m$ a positive integer larger than 1, then the cardinality of $Cl(1)$ modulo $n$ is $2m$.

Proof. From the lemma 15 we have $|Cl(1)| = \text{ord}_n q$. We want to prove that under the lemma’s hypothesis we have $\text{ord}_n q = 2m$. $q^{2m} - 1 = (q^m - 1)(q^m + 1)$ i.e., $q^{2m} \equiv 1 \mod (q^m + 1)$, so we have to prove that $2m$ is the smallest integer such that $q^{2m} \equiv 1 \mod (q^m + 1)$.

Assume there exists an integer $1 \leq r < 2m$ such that

$q^r \equiv 1 \mod q^m + 1$,

and then $r$ divides $2m$. Since we have $q^r = k(q^m + 1) + 1$, then $r > m$ and then $r$ is such that $m < r < 2m$ and $r$ divides $2m$. There is no integer which can verify these conditions.

Lemma 21 If $q$ is a prime power, $m \in \mathbb{N}$, $m > 1$ and $n = q^m + 1$, then all the cyclotomic classes

$Cl(s) = \{sq^l \mod n \mid l \in \mathbb{Z}\}$

with $s$ in the range $1 \leq s < q$, have the cardinality $|Cl(s)| = 2m$.

Proof. Let $1 \leq s < q$ and assume that $|Cl(s)| = i < 2m$, from lemma 17 we have that $|Cl(s)|$ divides $|Cl(1)|$ for all $s \in Z_n$, and then $i$ divides $2m$. Furthermore $i$ is the smallest integer which satisfies

$sq^i \equiv 1 \mod n \iff s(q^i - 1) \equiv 0 \mod (q^m + 1)$.
and then \( s \geq (q^m + 1)/(q^i - 1) \).

If \( i \leq m - 1 \), we have \( s \geq (q^m + 1)/(q^i - 1) \geq (q^m + 1)/(q^{m-1} - 1) > q \), which impossible since \( s < q \).

If \( i = m \), we have then \( sq^m = s + k(q^m + 1) \) and \( sq^m = -s + s(q^m + 1) \) by adding the two equalities we get

\[
2sq^m = (k + s)(q^m + 1).
\]

Since \( (q^m, q^m + 1) = 1 \), then we have \( 2s \equiv 0 \mod (q^m + 1) \), hence \( 2s \geq q^m + 1 \) which is impossible since \( s < q \) and \( m > 1 \).

Since \( i < 2m \) and \( i \) divides \( 2m \) there is no other case for \( i \).

**Lemma 22** For \( n = q^m + 1, m > 1 \) and \( s, s' \) distinct integers in the range \( 1 \leq s, s' < q \), we have the cyclotomic classes \( \text{Cl}(s) \) and \( \text{Cl}(s') \) are disjoint.

**Proof.** Assume there exists \( s \neq s' < q \) such that \( \text{Cl}(s) = \text{Cl}(s') \), and then there exist \( 1 \leq i < 2m \) and \( 1 \leq j < 2m \), such that \( s \equiv q^i s' \mod n \) and \( s' \equiv q^j s \mod n \) and \( i \neq 2m \) and \( i \neq 2m \), otherwise we will have \( s \equiv q^{2m} s' \mod n \), which implies \( s \equiv s' \mod n \). This is impossible since \( s, s' < q \). Now \( q^i s' \equiv s \mod n \iff q^i s' - s = k(q^m + 1), k \in \mathbb{N}^* \), because if \( k < 0 \) then, \( s - q^i s' = -k(q^m + 1) \), we will have then \( q > s = q^i s' - k(q^m + 1) \geq q^i + q^m + 1 \). For any value of \( i \) this is impossible. Consequently

\[
q^i s' - s = k(q^m + 1), \text{ with } k \in \mathbb{N}^*,
\]

hence this implies \( q^i s' \geq kq^m \), and then \( s' > q^{m-i} \), which is impossible for \( i \leq m - 1 \). By the same argument it is impossible to have \( j \leq m - 1 \).

For \( m \leq i \leq 2m - 2 \) we have:

\[
q^i s' = s + k(q^m + 1) \text{ and } q^m s = -s + s(q^m + 1),
\]

20
by adding the two equalities we get $q^m(s + q^{i-m}s') = (s + k)(q^m + 1)$, which implies $q^m$ divides $s + k$ and then

$$s + q^{i-m}s' \equiv 0 \mod (q^m + 1),$$

and then $s + q^{i-m}s' \geq q^m + 1$. Since $s' < q$ we will have

$$q > s > q^m - q^{i-m+1} + 1 \geq q^m - q^{m-1} + 1,$$

which is absurd.

By the same argument it is impossible to have $m \leq j \leq 2m - 2$.

If $i = 2m - 1$, $j = 2m - 1$, then we have

$$q^{2m-1}s' = s + k(q^m + 1).$$

(3.4)

By multiplying by $q$ the two members of the equality (3.4) we get $q^{2m}s' = sq + knq$, but $q^{2m} \equiv 1 \mod n$ so that we deduce that $s' \equiv sq \mod n$, and $s \equiv s'q \mod n$, and hence $s' \equiv s'q^2 \mod n$ and $s \equiv sq^2 \mod n$ i.e., $|Cl(s')| = 2$ and $|Cl(s)| = 2$, absurd. ■

### 3.3 Cyclic Codes

A linear code $C$ of length $n$ is called cyclic, if it is invariant under the cyclic shift

$$(a_1, \ldots, a_n) \mapsto (a_n, a_1, \ldots, a_{n-1}).$$

Consider the polynomial ring $\mathbb{F}_q[x]$ and its quotient ring

$$R_n = \mathbb{F}_q[x]/(x^n - 1),$$

if we ignore the multiplication we can construct a natural isomorphism of vector spaces.

$$\mathbb{F}_q^n \rightarrow \mathbb{F}_q[x]/(x^n - 1)$$
\[(a_0, \ldots, a_{n-1}) \mapsto a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}\]

The cyclic shift is just a multiplication by \(x\), and then a cyclic code \(C\) in \(R_n\) is just an ideal of \(R\). Since \(R_n\) is principal, then \(C\) is a principal ideal and we have the following properties of a cyclic code.

**Theorem 23** ([23] p190) Let \(C\) be a cyclic code of length \(n\) over \(\mathbb{F}_q\), then

1. There is a unique polynomial \(g(x)\) of minimal degree in \(C\) such that, \(C = \langle g(x) \rangle\) i.e., \(C\) is generated by the polynomial \(g(x)\).

2. \(g(x)\) is a divisor of \(x^n - 1\).

3. The dimension of \(C\) is \(n - r\), where \(r = \deg g\).

4. \(C = \{ f(x) \mod (x^n - 1) \mid f(a^i) = 0, \forall i \in T \}, \) where \(T\) is the *defining set* of \(C\), defined to be \(T = \cup \text{Cl}(j)\) the union is over all the cyclotomic classes \(\text{Cl}(j)\) such that \(M_j\) divides \(g\). Furthermore we have \(\deg g = |T|\).

**Theorem 24** ([23] p196) Let \(C\) be a cyclic code generated by the polynomial \(g\). Then the dual code \(C^\perp\) of \(C\) is a cyclic code generated by \(g^\perp = h^*\), the reciprocal polynomial of \(h(x) = \frac{x^n - 1}{g(x)}\), i.e., we have

\[g^\perp(x) = x^{\deg h} \cdot h(x^{-1})\]

### 3.4 BCH Codes

Let \(\alpha\) be a primitive \(n\)-th root of unity over \(\mathbb{F}_q\), a cyclic code \(C\) of length \(n\) over \(\mathbb{F}_q\) is called **BCH** (R.C. Bose, D.K. Ray-Chaudhuri and and A. Hocquenghem) with designed distance \(\delta \geq 1\) if its generator polynomial \(g\) satisfies:

\[\exists b \in \mathbb{N}, \quad g(a^b) = \cdots = g(a^{b+\delta-2}) = 0.\]
Thus $C$ is generated by $g(x) = \text{lcm}\{M_b(x), M_{b+1}(x), \ldots, M_{b+\delta-2}(x)\}$ and then the code is denoted by $B_q(n, \delta, b)$. We denote $B_q(n, \delta, 1)$ by $B_q(n, \delta)$ when $b = 1$ and it is called a narrow-sense BCH. When $\alpha$ is a primitive field element, from the equation (3.2) that is, when $n = q^s - 1$, then $B_q(n, \delta, \alpha, b)$ is referred to as a primitive BCH code. The primitive BCH code has been extensively studied in [23] and [30] and many others references.

In the next paragraph we will be interested in an “interesting non-primitive BCH code” using the terminology of [23] p268. We are particularly interested in giving dimensions and the minimum distance of a narrow-sense $B_q(n, \delta)$, code and its dual, $B_q^\perp(n, \delta)$, in the case when $n = q^m + 1$, and $m > 1$.

### 3.4.1 Dimension of BCH Codes

As a BCH code is a cyclic code then if it is generated by a polynomial $g$, from theorem 23 its dimension $k$ is equal to $n - \deg g$. Since the generator of a $B_q(n, \delta)$ is $g = \text{lcm}(M_1, \ldots, M_{\delta-1})$, often it is not easy to get the generator of $B_q(n, \delta)$, because we may have $M_i = M_j$ for $i \neq j$. In the next theorem we give this dimension, for a class of BCH codes.

**Theorem 25** Let $q$ be a prime power, and $m$ an integer larger than 1. Then a BCH code $B_q(n, \delta)$ of length $n = q^m + 1$ and designed distance $\delta$ in the range $2 \leq \delta \leq q$ has dimension,

$$k = q^m + 1 - 2m(\delta - 1).$$

**Proof.** From the definition of the minimal polynomial, we have that to each minimal polynomial $M_j$ of $\alpha^j$ there corresponds a cyclotomic class $Cl(j)$. Since from the Lemma 21 and the Lemma 22, the cyclotomic classes for $j < q$ have cardinality $2m$ and are disjoint, we can deduce that the minimal polynomials are distinct and each of degree $2m$. And hence the polynomial
generator of \( B_q(n, \delta) \) is
\[
g = \prod_{i=1}^{\delta-1} M_i,
\]
and its degree is \( 2m(\delta - 1) \). Therefore,
\[
\dim B_q(n, \delta) = q^m + 1 - 2m(\delta - 1).
\]

As the dual of an \([n, k]\) cyclic code is an \([n, n-k]\) cyclic code, the following corollary follows trivially from Theorem 25.

**Corollary 26** The dual code \( B_q(n, \delta) \perp \) has dimension
\[
k \perp = 2m(\delta - 1).
\]

### 3.4.2 The Minimum Distance

The problem of determining the exact value of the minimum distance remains unsolved for many families of codes, in particular the \( BCH \) codes. In [23], [30] or [18] and in many other references, under some conditions the problem is solved exactly or by giving upper and lower bounds. These bounds allow us to solve the problem not mathematically, but by finding physically the minimum codeword.

In the following paragraph we give an upper bound for the minimum distance of our \( BCH \) code.

For \( n = q^m + 1, \delta \leq q \), using the Singleton bound and Theorem 25 we obtain that
\[
\delta \leq d_{\text{min}} \leq 2m(\delta - 1) + 1. \quad (3.5)
\]
The following corollary of Theorem 25 will improve the bound (3.5).
**Corollary 27** If $B_q(n, \delta)$ is a BCH code of length $n = q^m + 1$ and designed distance $\delta \leq q$, such that

$$
\sum_{i=0}^{2(\delta-1)} \binom{q^m + 1}{i} (q - 1)^i > q^{2m(\delta-1)}, \quad (3.6)
$$

then we have

$$
d_{\text{min}} \leq 4(\delta - 1)
$$

**Proof.** Assume that $d_{\text{min}} \geq 4(\delta - 1) + 1$, from Theorem 25 the dimension is $q^m + 1 - 2m(\delta - 1)$. By the sphere-packing bound we have

$$
\sum_{i=0}^{\lfloor \frac{2(\delta-1)}{2} \rfloor} \binom{q^m + 1}{i} (q - 1)^i \leq q^{2m(\delta-1)},
$$

which implies that

$$
\sum_{i=0}^{2(\delta-1)} \binom{q^m + 1}{i} (q - 1)^i \leq q^{2m(\delta-1)},
$$

a contradiction. ■

To see the importance of corollary 27, we used the software Mathematica to find the values given in the next table and for which the corollary is verified, we compare the bound given by corollary 27 by the Griesmer bound and the Singleton bound. Our calculation have been done subject to the capacity of our computer, for that for many cases as shown by the tables the parameters verify the hypothesis (3.6) of the corollary 27 and then the bound, but we cannot compare the results to the Griesmer bound.
<table>
<thead>
<tr>
<th>$q$</th>
<th>$\delta$</th>
<th>$m$</th>
<th>The bound $4(\delta - 1)$</th>
<th>Griesmer bound</th>
<th>Singleton bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td>$d \leq 12$</td>
<td>$d \leq 10$</td>
<td>$d \leq 13$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 15$</td>
<td>$d \leq 19$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 20$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 24$</td>
<td>$d \leq 31$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 29$</td>
<td>$d \leq 37$</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 34$</td>
<td>$d \leq 43$</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 39$</td>
<td>$d \leq 49$</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq _________$</td>
<td>$d \leq 55$</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq _________$</td>
<td>$d \leq 61$</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
<td>hypothesis 3.6 not verified</td>
<td>$d \leq 20$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>$d \leq 16$</td>
<td>$d \leq 20$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 26$</td>
<td>$d \leq 33$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 33$</td>
<td>$d \leq 41$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 39$</td>
<td>$d \leq 49$</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 57$</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 65$</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 73$</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 81$</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 161$</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 241$</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 321$</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq __________$</td>
<td>$d \leq 401$</td>
</tr>
<tr>
<td>$q$</td>
<td>$\delta$</td>
<td>$m$</td>
<td>The bound $4(\delta - 1)$</td>
<td>Griesmer bound</td>
<td>Singleton bound</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>-----</td>
<td>--------------------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>$d \leq 8$</td>
<td>$d \leq 7$</td>
<td>$d \leq 9$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq 11$</td>
<td>$d \leq 13$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq 14$</td>
<td>$d \leq 17$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq 18$</td>
<td>$d \leq 21$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq 21$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq 25$</td>
<td>$d \leq 29$</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq$</td>
<td>$d \leq 33$</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq$</td>
<td>$d \leq 37$</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>$d \leq 8$</td>
<td>$d \leq$</td>
<td>$d \leq 41$</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>2</td>
<td>$d \leq 12$</td>
<td>$d \leq 11$</td>
<td>$d \leq 13$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 11$</td>
<td>$d \leq 19$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 16$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 21$</td>
<td>$d \leq 31$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 27$</td>
<td>$d \leq 37$</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq$</td>
<td>$d \leq 43$</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq$</td>
<td>$d \leq 49$</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq$</td>
<td>$d \leq 55$</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq$</td>
<td>$d \leq 61$</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2</td>
<td>$d \leq 16$</td>
<td>$d \leq 14$</td>
<td>$d \leq 17$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 14$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 21$</td>
<td>$d \leq 33$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 28$</td>
<td>$d \leq 41$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 35$</td>
<td>$d \leq 49$</td>
</tr>
<tr>
<td>( q )</td>
<td>( d )</td>
<td>( m )</td>
<td>The bound (4(\delta - 1))</td>
<td>Griesmer bound</td>
<td>Singleton bound</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>7</td>
<td>( d \leq 16 )</td>
<td>( d \leq )</td>
<td>( d \leq 57 )</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>( d \leq 16 )</td>
<td>( d \leq )</td>
<td>( d \leq 65 )</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>( d \leq 16 )</td>
<td>( d \leq )</td>
<td>( d \leq 73 )</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>( d \leq 16 )</td>
<td>( d \leq )</td>
<td>( d \leq 81 )</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>2</td>
<td>( d \leq 20 )</td>
<td>( d \leq 18 )</td>
<td>( d \leq 21 )</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq 27 )</td>
<td>( d \leq 31 )</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq 35 )</td>
<td>( d \leq 41 )</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq 44 )</td>
<td>( d \leq 51 )</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq 52 )</td>
<td>( d \leq 61 )</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq )</td>
<td>( d \leq 71 )</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq )</td>
<td>( d \leq 81 )</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq )</td>
<td>( d \leq 91 )</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>( d \leq 20 )</td>
<td>( d \leq )</td>
<td>( d \leq 101 )</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>2</td>
<td>( d \leq 8 )</td>
<td>( d \leq 8 )</td>
<td>( d \leq 9 )</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq 11 )</td>
<td>( d \leq 13 )</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq 15 )</td>
<td>( d \leq 17 )</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq 19 )</td>
<td>( d \leq 21 )</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq 10 )</td>
<td>( d \leq 25 )</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq )</td>
<td>( d \leq 29 )</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq )</td>
<td>( d \leq 33 )</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq )</td>
<td>( d \leq 37 )</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>( d \leq 8 )</td>
<td>( d \leq )</td>
<td>( d \leq 41 )</td>
</tr>
<tr>
<td>$q$</td>
<td>$\delta$</td>
<td>$m$</td>
<td>The bound $4(\delta - 1)$</td>
<td>Griesmer bound</td>
<td>Singleton bound</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>-----</td>
<td>---------------------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>2</td>
<td>$d \leq 12$</td>
<td>$d \leq 12$</td>
<td>$d \leq 13$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 17$</td>
<td>$d \leq 19$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 21$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 28$</td>
<td>$d \leq 31$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 37$</td>
<td>$d \leq 37$</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 43$</td>
<td>$d \leq 43$</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 49$</td>
<td>$d \leq 49$</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 55$</td>
<td>$d \leq 55$</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
<td>$d \leq 12$</td>
<td>$d \leq 61$</td>
<td>$d \leq 61$</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>2</td>
<td>$d \leq 16$</td>
<td>$d \leq 15$</td>
<td>$d \leq 17$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 22$</td>
<td>$d \leq 25$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 30$</td>
<td>$d \leq 33$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 37$</td>
<td>$d \leq 41$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 49$</td>
<td>$d \leq 49$</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>2</td>
<td>$d \leq 20$</td>
<td>$d \leq 19$</td>
<td>$d \leq 21$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq 28$</td>
<td>$d \leq 31$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq 37$</td>
<td>$d \leq 41$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq 46$</td>
<td>$d \leq 51$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq 61$</td>
<td>$d \leq 61$</td>
</tr>
</tbody>
</table>
The bound $4(\delta - 1)$

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\delta$</th>
<th>$m$</th>
<th>Griesmer bound</th>
<th>Singleton bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>5</td>
<td>2</td>
<td>$d \leq 16$</td>
<td>$d \leq 15$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 23$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 30$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq 38$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 16$</td>
<td>$d \leq$</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>2</td>
<td>$d \leq 20$</td>
<td>$d \leq 19$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq 28$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq 38$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td>$d \leq 20$</td>
<td>$d \leq$</td>
</tr>
</tbody>
</table>

### 3.4.3 Can our BCH Codes be Self-Orthogonal

A linear code $C$ is said to be self-orthogonal if it verifies

$$C \subset C^\perp.$$  

The following lemma, which is a result of the properties of the cyclotomic classes gives the link between the defining set of a cyclic code and its dual

**Lemma 28** Let $C$ be a cyclic code of length $n$ with defining set $T$ and $Z_n = \{0, \ldots, n-1\}$, then the defining set of its dual $C^\perp$ is

$$T^\perp = Z_n \setminus (-1)T,$$

where $(-)T = \cup CI(-i)$

**Proof.** [18] Theorem 4.4.9 ■
From the lemma 28 we can deduce that if all the cyclotomic classes are reversible then any cyclic code $C$ of length $n$ over $\mathbb{F}_q$ verifies that

$$C \cap C^\perp = \{0\}.$$  

And then the following corollary follows from lemma 19.

**Corollary 29** Any cyclic code of length $q^m + 1$ cannot be self-orthogonal.

Our $BCH$ codes are a particular case of the corollary 29.

### 3.5 Cyclic Reed-Solomon Codes

In this section we will define the Reed-Solomon codes as a subfamily of $BCH$ codes. A **cyclic narrow-sense Reed-Solomon Code**, $C$ over $\mathbb{F}_q$ is a narrow-sense $BCH$ code of length $n = q - 1$. Thus $\text{ord}_n q = 1$, implying that every cyclotomic class contains one element and consequently the degree of every minimal polynomial is 1.

Let $n = q - 1$, and choose $k$ in the range $1 \leq k \leq q - 1$, then the following polynomial

$$g(x) = \prod_{i=1}^{q-1-k} (x - \alpha^i)$$

is a generator of a cyclic code $C$ over $\mathbb{F}_q$ of length $q - 1$ and dimension $k$. Furthermore since we have $q - 1 - k$ consecutive roots, from the $BCH$ bound it follows that the minimal distance $d_{\text{min}}$ of $C$ is such that $d_{\text{min}} \geq q - k$. By the Singleton bound we have $d_{\text{min}} - k \leq q$ and then we have $d_{\text{min}} = q - k$ i.e., $C$ is $MDS$. This definition of $RS$ codes is totally different from the definition given in section 2.3. The following theorem gives the link between the two definitions. As in section 2.3, for $k \in \mathbb{N}$, we consider the vector space

$$\mathcal{L}_k = \{f(x) \in \mathbb{F}_q[x] \mid \deg f \leq k - 1\}.$$
Theorem 30 ([18] Theorem 5.2.3) Let $\alpha$ be an $n^{th}$ root of unity and let $k$ be an integer with $0 \leq k \leq n = q - 1$. Then

$$C = \{(f(1), f(\alpha), \ldots, f(\alpha^{q-2})) \mid f \in \mathcal{L}_k\}$$

is the narrow-sense $[n, k, n - k + 1]$ RS code over $\mathbb{F}_q$.

**Proof.** We consider the following linear map

$$\text{ev} : \mathcal{L}_k \longrightarrow \mathbb{F}_q^n,$$

$$f \longmapsto (f(1), f(\alpha), \ldots, f(\alpha^{q-2})). \tag{3.7}$$

The code $C$ is the image of the linear map $\text{ev}$ and so is a linear code. Since $k \leq n$ a non zero polynomial $f(x) \in \mathcal{L}_k$ cannot vanish at all $\alpha^i$. Hence the evaluation map $\text{ev}$ is injective, and it follows therefore that $\dim C = \dim \mathcal{L}_k = k$. Let $D$ be a narrow sense $[n, k, n - k] \text{ RS code over } \mathbb{F}_q$. So $D$ has defining set $T = \{1, 2, \ldots, n - k\}$, and to show that $C = D$, it suffices to show that $C \subset D$ as both are $k$-dimensional. Let $c(x) = \sum_{j=0}^{n-1} c_i x^i \in C$. Then there exist some $f(x) = \sum_{m=0}^{k-1} f_m x^m \in \mathcal{L}_k$ such that $c_j = f(\alpha^j)$ for $0 \leq j < n$. To show that $c(x) \in D$ we need to show that $c(\alpha^i) = 0$ for $i \in T$, by theorem 23. If $i \in T$, then

$$c(\alpha^i) = \sum_{j=0}^{n-1} c_j \alpha^{ij} = \sum_{j=0}^{n-1} \left(\sum_{m=0}^{k-1} f_m \alpha^{jm}\right) \alpha^{ij} = \sum_{m=0}^{k-1} f_m \sum_{j=0}^{n-1} \alpha^{(i+m)j} = \sum_{m=0}^{k-1} f_m \frac{\alpha^{(i+m)n} - 1}{\alpha^{i+m} - 1}.$$ 

Now $\alpha^{(i+m)n} = 1$ and $\alpha^{i+m} \neq 1$ as $1 \leq i + m \leq n - 1 = q - 2$ and $\alpha$ is a primitive $n^{th}$ root of unity. Therefore $c(\alpha^i) = 0$ for $i \in T$, implying that $C \subset D$. Hence $C = D$. ■
Chapter 4

The List-Decoding Algorithm for Reed–Solomon Codes

The list-decoding is considered as a relaxation of the unique decoding and allows the decoder to output a list of codewords as answer. The idea of the list-decoding was first introduced by Elias [9] in 1957, but it had no important algorithm until 1997 when Sudan, building on the previous work of Ar et al [2], gave an efficient algorithm, which allows one to correct beyond the minimum distance. The Berlekamp–Welch decoding algorithm which uses $O(n^3)$ operations fields was improved by Justesen in [20] to work with $O(npoly \log n)$. Even this improvement it is much better to use the list-decoding algorithm, since for $R \to 0$ it can correct an error rate $\tau/n$ which approaches 1. This allows for nearly twice as many errors as the classical approach. For a rate greater than 1/3, this algorithm does not improve over the classical approach.

The principle of the list-decoding is the following. From Lemma 6, we have seen that it may not be possible to uniquely reconstruct a codeword from a received word if the number of errors is beyond the bound $\frac{d-1}{2}$, and hence the necessity to return a set of candidates which probably includes the
original codewords. See Figure 4.1

Figure 4.1: If we want to correct $\tau$ errors beyond $(d - 1)/2$, the Hamming ball of radius $\tau$ contains $l \geq 1$ codewords.

This motivates the following definition.

**Definition 31** Given an $[n, k, d]$-code $C$, we say that $C$ is a $(\tau, l)$-list decodable if for any received word $r \in \mathbb{F}_q^n$ the Hamming ball $B_\tau(r)$ of radius $\tau$ and centre $r$ contains at most $l$ codewords, i.e.

$$\forall r \in \mathbb{F}_q^n, |B_\tau(r) \cap C| \leq l.$$ 

*Note the volume of the Hamming ball $B_\tau(r)$ is $\sum_{i=0}^{\tau} \binom{n}{i}(q - 1)^i$. The problem of finding $x$ in $B_\tau(r)$ is to find $x \in \mathbb{F}_q^n$ such that $x$ differs from $r$ in at most $\tau$ positions.*

This problem is equivalent to find $x \in \mathbb{F}_q^n$ such that

$$x_i = r_i \text{ for at least } n - \tau \text{ values of } i.$$
For a Reed–Solomon code, suppose $c = (f(α_1), \ldots, f(α_n))$ is a transmitted codeword, where $f \in \mathcal{L}_k$ (the set of polynomial over $\mathbb{F}_q$ of degree less than $k$), and that $c$ is received as $r = (r_1, \ldots, r_n)$. To decode $r$ by a list of length $l$ is to find a list of $x = (p(α_1), \ldots, p(α_n))$, where $p(x)$ is any polynomial of degree less than $k$, such that $p(α_i) = r_i$ for at least $n − τ$ values of $i$. The set of such polynomials is denoted by $B^*_τ(r)$. Then the problem of finding the set $B_τ(r)$ of length $l$ is replaced by the problem of finding the set $B^*_τ(r)$ of all polynomials $p \in \mathcal{L}_k$, such that

$$p(α_i) = r_i \text{ for at least } n − τ \text{ values of } i.$$ 

We have

$$p \in B^*_τ(r) \iff (p(α_1), \ldots, p(α_n)) \in B_τ(r).$$

It is obvious that the balls $B^*_τ(r)$ and $B_τ(r)$ have the same size. Then the problem of list-decoding of the Reed–Solomon code is the following:

**Given:** A set of $n$ distinct pairs $\{(α_i, r_i) \in \mathbb{F}_q \times \mathbb{F}_q \mid i \in [n]\}$ and integers $τ$ and $k$.

**Find:** A list of all polynomials $p$ of degree at most $k$ satisfying $p(α_i) = r_i$ for at least $n − τ$ values of $i \in [n]$.

### 4.1 Sudan’s Algorithm for Reed Solomon codes

The list-decoding presented here is the list-decoding with constant sized list. For the analysis on the parameters we follow the presentation given in [19], which was based on [39]. We adapt it with a discussion on the bounds on the parameters as given in [38] and [14].

Before we state Sudan’s theorem, we need to recall the definition of the weighted degree of a bivariate polynomial.
Definition 32  Consider the bivariate polynomial:

\[ Q(x, y) = \sum_{i,j} q_{i,j} x^i y^j. \]

Let \( w_1, w_2 \) be non-zero integers. The \((w_1, w_2)\)-weighted degree of \( Q(x, y) \) is defined as:

\[ \text{wdeg}_{(w_1, w_2)} Q(x, y) = \max \{ iw_1 + jw_2 \mid q_{i,j} \neq 0 \}. \]

The idea of Sudan is to determine a non-zero polynomial

\[ Q(x, y) = Q_0(x) + Q_1(x)y + \ldots + Q_l(x)y^l \]

satisfying the hypothesis of the following theorem.

Theorem 33  Let \( Q(x, y) \in F_q[x, y] \). If \( Q(x, y) \neq 0 \), satisfies

1. \( Q(\alpha_i, r_i) = 0 \), for all \( i \in [n] \), and
2. \( \text{wdeg}_{(1, k-1)} Q(x, y) < n - \tau \),

then \((y - p(x))\) divides \( Q(x, y)\) for all \( p \in B^*_r(r) \)

Proof. Let \( p \) be a polynomial in \( B^*_r(r) \), and consider \( Q_p(x) \) to be the polynomial \( Q(x, p(x)) \). Since \( \deg p < k \), and \( \text{wdeg}_{(1, k-1)} Q(x, y) < n - \tau \) then \( \deg Q_p(x) < n - \tau \). Moreover \( p(\alpha_i) = r_i \) for at least \( n - \tau \) values of \( i \), so \( Q_p(x) \) has at least \( n - \tau \) roots. Hence \( Q_p(x) \) is identically zero, i.e., \((y - p(x))\) divides \( Q(x, y) \).

A polynomial \( p(x) \) in \( \mathcal{L}_k \) is called a \textbf{y-root} of \( Q(x, y) \) if it verifies

\[ y - p(x) \mid Q(x, y). \]
Remark 34 The condition given by theorem 33 is a necessary condition, we can have $p(x)$ is a $y$-root but it is not in $B_r(x, y)$. To be in $B_r(x, y)$ we have to verify that $p(\alpha_i) = r_i$ for at least $n - \tau$ values.

The polynomial $Q$ as in Theorem 33 (depending on $\tau$) is called an interpolation polynomial.

Algorithm 1: List-decoding of RS codes using the Sudan Algorithm

Input: a received word $r = (r_1, \ldots, r_n)$ and a natural number $l$.

Step S1 (Interpolation) : Find $Q(x, y) = \sum_{j=0}^{l_j} \sum_{s=0}^{l_j} q_{j,s} x^s y^j$ as in Theorem 33; this step requires solving the following linear system of unknowns,

$$
\begin{pmatrix}
  r_1^j & \cdots & 0 & 0 \\
  0 & r_2^j & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & r_n^j
\end{pmatrix}
\begin{pmatrix}
  1 & \alpha_1 & \alpha_1^{l_1} \\
  1 & \alpha_2 & \alpha_2^{l_2} \\
  \vdots & \vdots & \ddots \\
  1 & \alpha_n & \alpha_n^{l_n}
\end{pmatrix}
\begin{pmatrix}
  q_{j,0} \\
  q_{j,1} \\
  \vdots \\
  q_{j,l_j}
\end{pmatrix} =
\begin{pmatrix}
  0 \\
  0 \\
  \vdots \\
  0
\end{pmatrix}
$$

where $l_j = n - \tau - j(k - 1) - 1$.

Put the result in

$$Q_j(x) = \sum_{s=0}^{l_j} q_{j,s} x^s.$$

Step S2 (Factorisation) : Factorise $Q(x, y)$ into irreducible factors in $\mathbb{F}_q[x, y]$.

Get all factors of the form $(y - p(x))$ and keep only the $p$'s for which $p(\alpha_i) = r_i$ for at least $n - \tau$ values of $i$.

Output: A list of $l$ codewords $c = (p(\alpha_1), \ldots, p(\alpha_n))$ obtained from the factors $p(x)$ above, that satisfy $d(c, r) \leq \tau$. 

37
Remark 35 The interpolation polynomial is not unique since the solution of the system is not unique.

We will discuss under which conditions on $\tau$ and $l$ such an interpolating polynomial exists.

The number of unknowns is determined by condition (2) in Theorem 33. Since this condition is equivalent to the following

$$\deg (Q_j(x)) \leq n - \tau - 1 - j(k - 1), \text{ for } j = 0, \ldots, l$$

and because

$$wdeg_{(1,k-1)} Q(x,y) < n - \tau$$
$$\Leftrightarrow \max (\deg Q_j + (k - 1)j) < n - \tau$$
$$\Leftrightarrow \deg Q_j \leq n - \tau - (k - 1)j - 1 \text{ for } j = 0, \ldots, l$$

the number of unknowns is

$$(n - \tau) + (n - \tau - (k - 1)) + (n - \tau - 2(k - 1)) + \ldots + (n - \tau - l(k - 1))$$
$$= (l + 1)(n - \tau) - \frac{1}{2}l(l + 1)(k - 1).$$

By condition (1) in the definition of $Q(x,y)$, there are $n$ linear equations, so there are more unknowns than constraining equations provided,

$$(l + 1)(n - \tau) - \frac{1}{2}l(l + 1)(k - 1) > n.$$ 

This is equivalent to the following inequality

$$\tau < \frac{nl}{l + 1} - \frac{l(k - 1)}{2}. \quad (4.1)$$
If \( l = 1 \), we get
\[
\tau < \frac{n - k + 1}{2} = d/2
\]
and this is not interesting, because we want \( \tau > d/2 \). If \( l \geq 2 \), and since \( \tau > 0 \), we get
\[
n > \frac{(l + 1)(k - 1)}{2} \geq \frac{2}{3}(k - 1).
\]
Hence, from (4.1) for \( l = 2 \), we get
\[
\frac{\tau}{n} < \frac{2}{3} + \frac{1}{n} - \frac{k}{n},
\]
which is impossible for \( R = \frac{k}{n} \geq \frac{2}{3} + \frac{1}{n} \).

For \( \frac{k}{n} < \frac{2}{3} + \frac{1}{n} \), we only get an improvement on half the minimum distance if
\[
\frac{k}{n} < \frac{1}{3} + \frac{1}{n},
\]
because we want
\[
\frac{n - k + 1}{2} < \tau < \frac{2}{3}n - (k - 1) \Leftrightarrow \frac{k}{n} < \frac{1}{3} + \frac{1}{n}.
\]

For the problem of list-decoding we must have
\[
\tau > \frac{n - k}{2}. \quad (4.2)
\]

If we combine conditions (4.1) and (4.2), we obtain the following inequalities, which are equivalent,
\[
\frac{n - k}{2} < \tau < \frac{nl}{l + 1} - \frac{l(k - 1)}{2},
\]
\[
\frac{2n - (l + 1)(k - 1)}{2(l + 1)} > n - k.
\]
\[
\frac{k - \frac{l}{k-1}}{n} < R < \frac{1}{l+1} + \frac{l}{(l-1)n}.
\]

From condition (2) and for \(j = l\), we have

\[
0 \leq \deg Q_l(x) \leq n - \tau - 1 - l(k-1).
\]

This implies \(\tau \leq n - 1 - l(k-1)\), or equivalently \(l < \frac{n-1-\tau}{k-1}\). And then we have the rough bound on \(l\)

\[
l < \frac{n}{k-1} \tag{4.3}
\]

Now the list-decoding applies when the code has low rate, and when

\[
\frac{n-k}{2} < \tau < n(1 - Rl) + l - 1.
\]

**Example 36** Decoding the Reed–Solomon code \([31, 5]\) over \(F_{32}\) will give the following for the length and the errors corrected.

- \(l = 1\) gives \(\tau \leq 13\).
- \(l = 2\) gives \(\tau \leq 16\).
- \(l = 3\) gives \(\tau \leq 18\).
- \(l = 4\) gives \(\tau \leq 16\).

We remark that the optimal length is 3.

**Example 37** For the \([10, 3]\) Reed–Solomon over \(F_{11}\). If we want to decode the received word \((0, 0, 6, 9, 1, 6, 0, 0, 0, 0)\) with a list \(l\) of length 2 we can correct 4 errors, the conventional algorithms can correct up to 3 errors. The polynomial \(Q(x, y) = y(y - (x^2 - 3x + 2))\) can be found by the Sudan algo-
rithm, and then the possible codewords are

$$(0, 0, 0, 0, 0, 0, 0, 0, 0)$$ and $$(0, 0, 4, 8, 1, 6, 1, 8, 2, 9)$$. 

### 4.2 Guruswami–Sudan Algorithm

Since many applications tend to work with codes of high rates (even if the Hadamard codes which are of low rate were used by Mariner [43] p 26), there was a need to find a list-decoding algorithm for the highest rates. For that the Sudan’s algorithm was extended to the **Guruswami–Sudan’s algorithm**. The extended algorithm has the same steps as the Sudan’s algorithm but works for highest rates. Before giving the algorithm we need some definitions.

The shifting polynomial for the polynomial $Q(x, y)$ and $(a, b) \in \mathbb{F}_q \times \mathbb{F}_q$ is defined by

$$Q_{a,b}(x, y) = Q(x + a, y + b)$$

This gives us the following explicit relation between the coefficients $\{q_{ij}\}$ and the coefficients $\{q^*_{ij}\}$ of $Q_{a,b}$

$$q^*_{ij} = \sum_{i' \geq i} \sum_{j' \geq j} \binom{i'}{i} \binom{j'}{j} q_{i', j'} a^{i'-i} b^{j'-j}.$$ 

This is a consequence of the binomial theorem. We express $Q(x + a, y + b)$
We remark also that the shifting does not change the weighted degree.

The next definition leads to the main difference between Sudan’s algorithm and Guruswami–Sudan’s algorithm.

**Definition 38** We say that a polynomial $Q(x, y)$ in $\mathbb{F}_q[x, y]$ has a zero of multiplicity $s$ at $(0, 0)$, if $Q(x, y)$ has no term of total degree less than $s$, i.e., $q_{ij} = 0$ if $i + j < s$. Similarly, we say that $Q(x, y)$ has a zero of multiplicity $s$ at $(a, b)$ or that $Q(x, y)$ passes through the point $(a, b)$ at least $s$ times if $Q(x + a, y + b)$ has a zero of multiplicity $s$ at $(0, 0)$, i.e.,

$$q^*_{i,j} = 0 \text{ for } i + j < s,$$

where $q^*_{ij}$ are the coefficients of the polynomial $Q_{a,b}$.

There is no limit on the multiplicity $s$, only the running time increase with $s$.

The second algorithm has a similar step to the first algorithm, but it requires more from $Q$. It finds a non-zero polynomial of low degree

$$Q(x, y) = Q_0(x) + Q_1(x)y + \ldots + Q_l(x)y^l$$

such that:
1. \((\alpha_i, r_i)\) for \(i \in [n]\) are zeroes of multiplicity \(s\) of \(Q(x,y)\),

2. \(\deg (Q_j(x)) \leq s(n - \tau) - 1 - j(k - 1)\), for \(j = 0, 1, \ldots, l\).

We then have

**Lemma 39** If \(Q(x,y)\) satisfies the above conditions and \(c = (p(\alpha_1), \ldots, p(\alpha_n))\) with \(\deg (p(x)) < k\), then \((y - p(x)) \mid Q(x,y)\)

**Proof.** We first prove the following claim: if \(p(\alpha_i) = r_i\), then \((x - \alpha_i)^s \mid Q(x,p(x))\).

For that, let \(P(x) = p(x + \alpha_i) - r_i\). We then have \(P(0) = 0\) and therefore \(x \mid P(x)\).

If \(h(x) = Q(x + \alpha_i, P(x) + \alpha_i)\), it follows from condition (1) that \(h(x)\) has no terms of degree smaller than \(s\) so \(x^s \mid h(x)\) and therefore \((x - \alpha_i)^s \mid h(x - \alpha_i)\).

Since \(h(x - \alpha_i) = Q(x,p(x))\), the claim is proved. The proof of the lemma follows by noting that the polynomial \(Q(x,p(x))\) has degree at most \(s(n - \tau) - 1\).

But \((x - \alpha_i)^s \mid Q(x,p(x))\) for at least \(n - \tau\) of the \(\alpha_i\)'s and so \(Q(x,p(x))\) is divisible by a polynomial of degree at least \(s(n - \tau)\) and hence \(Q(x,p(x)) = 0\).

Just as for algorithm 1, this implies that we can find all codewords of distance at most \(\tau\) from the received word by finding the \(y\)-roots of \(Q(x,y)\) with \(\deg (p(x)) < k\).

Now we will discuss the condition on \(\tau\). The polynomial \(Q(x,y)\) must satisfy condition (2) which gives non-negative degrees if

\[
s(n - \tau) - l(k - 1) \geq 0, \tag{4.4}\]

or equivalently

\[
\tau \leq n - \frac{l(k - 1)}{s}. \tag{4.5}\]

The first condition is a system of \(n\binom{s+1}{2}\) homogeneous linear equations, because \(a_{i,j}^* = 0\) for \(i + j < s\). Then, if the number of unknowns is greater
than this number, the system has non-zero solutions. Condition (2) gives
the number of unknowns as

\[ s(n - \tau) + s(n - \tau) - (k - 1) + \ldots + s(n - \tau - l(k - 1) \]
\[ = (l + 1)s(n - \tau) - \frac{1}{2}l(l + 1)(k - 1). \]

From this the condition is

\[ (l + 1)s(n - \tau) - \frac{1}{2}l(l + 1)(k - 1) > n \left( \binom{s + 1}{2} \right), \]

or equivalently

\[ \tau < \frac{n(2l - s + 1)}{2(l + 1)} - \frac{l(k - 1)}{2s}. \] (4.6)

If we want to get an improvement on half the minimum distance we should have

\[ \frac{d - 1}{2} = \frac{n - k}{2} < \tau < \frac{n(2l - s + 1)}{2(l + 1)} - \frac{l(k - 1)}{2s} \] (4.7)

or equivalently

\[ k \frac{(l - s)}{s} < n \frac{(l - s)}{l + 1} + \frac{l}{s}. \] (4.8)

The above discussion implies that an \([n, k] \) RS code is \((\tau, l)\) decodable as long as \(\tau\) verifies the equation (4.8).

**Remark 40** We remark that we cannot have \(s \geq l + 1\), for then the equation (4.7) will imply the following

\[ k > n + \frac{1}{l - s}, \]

which is absurd, because then \(k + 1 > n\).
By setting $l > s$ in the equation (4.8), we get

$$\frac{k}{n} < \frac{s}{l + 1} + \frac{l}{n(l - s)} \leq \frac{s}{l + 1} + \frac{1}{n}.$$ 

For $s = 1$ we get, $l < n/(k - 1)$ which is the same bound as given in the equation (4.3).

We now formulate the algorithm.

**Algorithm 2 :** List-decoding for RS codes using the Guruswami–Sudan Algorithm.

**Input:** A received word $(r_1, \ldots, r_n)$

**Parameters:** $n, k, \tau$

**Step 0** Compute parameters $s, l$ such that

$$\tau \leq n - \frac{l(k - 1)}{s}$$

and

$$\tau < \frac{n(2l - s + 1)}{2(l + 1)} - \frac{l(k - 1)}{2s}.$$

**Step S1 (Interpolation):** Find a polynomial $Q(x, y)$ such that

$$\deg (Q_j) \leq s(n - \tau) - 1 - j(k - 1), j = 0, 1, \ldots, l$$

i.e., find values for its coefficients

$$\{q_{j_1, j_2}\}_{j_1, j_2 \geq 0; j_1 + j_2 \leq l},$$

such that the following conditions hold:

1. At least one $q_{j_1, j_2}$ is nonzero.

2. For every $i \in [n]$, if $Q^{(i)}$ is the shift of $Q$ to $(\alpha_i, r_i)$, then all coefficients
of $Q^{(i)}$ of total degree less than $s$ are 0. More specifically

$$\forall i \in [n], \forall j_1, j_2 \geq 0, \text{ s.t. } j_1 + j_2 < s$$

$$q_{j_1, j_2}^i = \sum_{j_1' \geq j_1} \sum_{j_2' \geq j_2} \binom{j_1'}{j_1} \binom{j_2'}{j_2} q_{j_1', j_2'}^{i} \alpha_{i}^{j_1' - j_1} r_{i}^{j_2' - j_2} = 0.$$  

3. Solve for $Q^{(i)}$ in the system of linear equations, $\forall i \in [n], \forall j_1, j_2 \geq 0$, s.t. $j_1 + j_2 < s$:

$$q_{j_1, j_2}^i = \sum_{j_1' \geq j_1} \sum_{j_2' \geq j_2} \binom{j_1'}{j_1} \binom{j_2'}{j_2} q_{j_1', j_2'}^{i} \alpha_{i}^{j_1' - j_1} r_{i}^{j_2' - j_2} = 0,$$

with $q_{j_1', j_2'}^{i} = 0$ if $l > j_1'$ or $j_2' > l_{j'}$ where $l_{j'} = s(n - \tau) - 1 - j'(k - 1)$.

4. Put

$$Q_j(x) = \sum_{t=0}^{l_j} q_{j,s} x^t \text{ and } Q(x,y) = \sum_{j=0}^{l} Q_j(x) y^j.$$  

Step S2 (Factorisation): Find all factors of $Q(x, y)$ of the form $(y - p(x))$ with $deg\ (p(x)) < k$.

Output: A list of factors $p(x)$ that satisfy

$$d(c, r) < \tau.$$  

4.2.1 The Asymptotic Bound on the Radius

From the equation (4.7) we have that an $[n, k]$ $RS$-code is $(\tau, l)$-list-decodable as long as

$$\tau < \frac{n(2l - s + 1)}{2(l + 1)} - \frac{l(k - 1)}{2s}.$$  

(4.9)
For \( \tau = \left\lfloor \frac{n(2l-s+1)}{2(l+1)} - \frac{l(k-1)}{2s} \right\rfloor \) we get that

\[
s(n-\tau) = \frac{l(k-1)}{2} + \frac{sn(1+s)}{2(l+1)}.
\]

From the equation (4.4) the parameters must satisfy \( s(n-\tau) \geq l(k-1) \). If we combine this with equation (4.10) we find that

\[
\frac{s(s+1)}{l(l+1)} \geq \frac{k-1}{n},
\]

which gives \( \frac{s^2}{l^2} \approx \frac{k}{n} \), and then when \( l \to \infty \), if we divide by \( n \) the inequality (4.9) we get the following

\[
\frac{\tau}{n} < 1 - \sqrt{k/n} = 1 - \sqrt{R}.
\]

The bound given in the inequality (4.11) is called the **Johnson bound for Reed–Solomon codes**.

The conventional decoding algorithm gives the bound \( \frac{\tau}{n} \leq \frac{1-R}{2} \). To see the difference, let us consider the extremal case, when we let the rate approach zero:

\[
\lim_{R \to 0} 1 - \sqrt{R} = 1 \quad (\text{so } \tau \to n)
\]

\[
\lim_{R \to 0} \frac{1 - \sqrt{R}}{2} = 1/2 \quad (\text{so } \tau \to n/2)
\]

The above equations prove that theoretically when the rate approaches zero the list-decoding can correct up to \( n \) errors i.e., 100\% of the errors, nevertheless the conventional decoding algorithms only correct up to 50\% errors.\(^1\)

\(^1\)In [15] it was conjectured that asymptotically \( 1 - \sqrt{R} \) is the largest fraction of error that can be corrected, more recently in [8] it was proved that the list-decoding cannot be done for the radius \( n - g'(n, k, q) \) or larger where \( g'(n, k, q) \) is the smallest integer \( g \) such that \( \binom{n}{g} < 1 \), otherwise the discrete logarithm over \( \mathbb{F}_{q^{g'(n,k,q)}} \) would become easy.
4.2.2 The Run time of the Algorithm

When Sudan gave the algorithm in [38], he only mentioned that the algorithm can be implemented in polynomial time, since the resolution of the system of unknowns and the factorization of the interpolation polynomial can be done in polynomial time. Later Nielsen and Hoholdt [28] gave a solution to the interpolation step, which can be implemented in $O(n^2s^5)$ operations. Olshevsky and Shokrollahi in [29] provide a different implementation using $O(n^2s^4\log_q s)$ operations, and that by using the “displacement method” applied to find non-zero elements in the kernel of certain structures matrices. The factorization step and its cost will be seen in the next chapter.
Chapter 5

The Factorization Step in the List-Decoding Algorithm

An efficient solution to the factorization step in the Sudan algorithm based on the Hensel lifting and the Newton approximation of a root, was proposed by Lancelot Pecquet and Daniel Augot in [3]. The algorithm use $O(n^2 \log n)$ operations using the Fast Fourier Transformation, but the method failed to be generalized to Guruswami-Sudan algorithm. In this chapter we propose a solution to the factorization step which can be applied to both algorithms, and which use in average

$$O\left(\frac{m}{2}(m'M(l) + lM(m')) + m^2\right) \log q)$$

field operations, with $m$ the degree of the first nonzero coefficient and $m' = \deg_x Q(x,y)$. 
5.1 The Factorization of the Interpolation Polynomial

The following method was given in [19] as a root-finding algorithm, solution to the factorization step in the Guruswami–Sudan algorithm. It is based on the simple fact that we have to find an irreducible polynomial $h(x)$ of degree $k$ over $\mathbb{F}_q$ and using this $h(x)$, we construct $E = \mathbb{F}_{q^k}$. For a polynomial $p(x)$, let $[p(x)]$ denote the element in $E$ corresponding to $p(x) \mod h(x)$. Let $\Phi$ be the map

$$
\Phi : \mathbb{F}_q[x, y] \rightarrow E[y]
$$

$$
\sum_i p_i(x)y^i \mapsto \Phi(\sum_i p_i(x)y^i) = \sum_i [p_i(x)]y^i.
$$

(5.1)

So then the effect of the map $\Phi$ is to reduce the $p_i$ modulo $h(x)$ and then $\Phi(Q(x, y))$ can be considered as an element of $\mathbb{F}_{q^k}[y]$. By using the fact that $\Phi(Q_1Q_2) = \Phi(Q_1)\Phi(Q_2)$ and $\Phi(Q_1 + Q_2) = (\Phi(Q_1) + \Phi(Q_2)$ we have the following lemma.

**Lemma 41** If $(y - f(x)) \mid Q(x, y)$, then $y - [f]$ is a factor of $\Phi(Q(x, y))$.

This means that the factorization problem is reduced to factoring a univariate polynomial.

The cost of the algorithm above is the total of the cost of each step.

1. Even if the existence of the extension field $\mathbb{F}_{q^k}$ known by a theorem due to Moore (1893), the construction can be done by finding the irreducible polynomial $h(x)$ of degree $k$ in $O((k \log k + \log q)M(k))$ field operations by the Shoup’s algorithm given in [33].

2. To reduce each polynomial $Q_i(x)$ of degree less than $m'$ modulo $h(x)$ can be done with an expected number of $O(m'k)$ by the extended Eu-
clidean algorithm (Theorem 3.11 given by Von Zur Gathen and Gerhard in [44]. Since there are \( l \) polynomials the cost of this step is \( O(m'n) \) operations.

3. The factorization of the polynomial \( \Phi(Q(x, y)) \) which is a polynomial of degree \( l \) over \( E = \mathbb{F}_{q^k} \) can be done by an expected number of \( O((l^{1.815})k \log q) \) operations, i.e., \( O(nl \log q) \) operations, (since \( lk < n \)).

From the previous analysis, we get that the total cost of the algorithm root-finding is

\[
O((k \log k + \log q)M(k) + m'n + nl \log q).
\]

### 5.2 A Solution to the Factorization Step in the List-Decoding Algorithm

In the following we propose a solution to the factorization step which can be applied to both algorithms, and which runs in time

\[
O(m/2(m'M(l) + lM(m')) + m^2) \log q),
\]

with \( m \) the degree of \( Q_0(x) \) the first nonzero coefficient of the interpolation polynomial \( Q(x, y) \) assumed to be monic, and \( m' = \text{deg}_y Q(x, y) \).

The method is based on an important property of the polynomials \( f \) \( y \)-roots. Even though this property which lead to the algorithm proposed in this chapter, is based on very simple idea, the analysis of the algorithm shows its importance.

The research of other properties should yield an efficient algorithm. The property of the \( y \)-roots is given in the following lemma.

**Lemma 42** Let \( Q(x, y) \) be the interpolation polynomial given by the interpolation step, then the following holds
1. if $Q_0(x) = 0$, there exist a null $y$-root.

2. the $y$-roots divide the first nonzero coefficient of $Q(x, y)$.

**Proof.** By construction of the interpolation polynomial

$$Q(x, y) = Q_0(x) + Q_1(x)y \ldots + Q_l(x)y^l,$$

the existence of the factors of the form $y - f_i(x)$ is assured for $1 \leq i \leq t$ for some $t \leq l$ and that from theorem 33. If $Q_{m_0}(x)$ is the first non-zero coefficient, then

$$Q(x, y) = y^{m_0}Q_{m_0}(x) + \ldots + y^lQ_l(x) = y^{m_0}(Q_{m_0}(x) + \ldots + y^{l-m_0}Q_l(x)).$$

The factorization of $Q(x, y)$ is then

$$Q(x, y) = y^{m_0}(y - f_1(x)) \ldots (y - f_t(x))P(x, y), \quad (5.2)$$

with $P(x, y) = P_0(x) + yP_1(x, y)$, and $P_0(x)$ not identically zero, because if $P_0(x) = 0$ that implies that $Q_{m_0+1}(x)$ is the first non-zero coefficient. Let $M(x, y)$ be the polynomial equal to $M(x, y) = Q_{m_0}(x) + \ldots + y^{l-m_0}Q_l(x)$, which from (5.2) is equal to

$$M(x, y) = (y - f_1(x)) \ldots (y - f_t(x))P(x, y), \quad (5.3)$$

since we are in a unique factorization domain. The way to write $M(x, y)$ is to consider the form (5.3) as a polynomial in $y$ and then the constant term in (5.3), which is $(-1)^tP_0(x)\prod_{i=1}^{t} f_i(x)$ is equal to $Q_{m_0}(x)$, hence the result. ■

**Conclusion 43** The previous property of the $y$-roots suggests to us to get the $y$-roots of the interpolation polynomial $Q(x, y)$, we have to apply the algorithm...
3 if \( k \leq m/2 \) or the similar algorithm 3’ in the other case. The polynomial \( Q_0(x) \) with degree \( m < s(n - \tau) \), denotes the first nonzero coefficient of \( Q(x, y) \). We will find the average running time of the algorithm under the assumption that \( Q_0 \) is monic.

Algorithm 3

**Input:** The polynomial \( Q_0(x) \) with \( m = \deg Q_0 \).

**Parameters:** integer \( k \) such that \( k \leq m/2 \).

1. **Step \( F_1 \)**
   - **Factorise** the monic univariate polynomials \( Q_0(x) \), list all the pairs \( \{f, h\} \) of monic factors such that \( Q_0(x) = f(x).h(x) \), with \( \deg f \leq \deg h \) and \( \deg f \leq m/2 \).
   - Define \( C_2 = \{(f(x), h(x)) \mid 1 \leq \deg f < k \leq \deg h \} \)

2. **Step \( F_2 \)**
   - \( i = 1 \)
   - For \( \{f, h\} \in C_2 \)
     - If \((y + f(x))|Q(x, y)\)
       - \( Z_i = -f(x) \).
     - If \( i < l \)
       - \( i = i + 1 \)
     - Otherwise
       - **Output:** \( Z_1, \ldots, Z_N \) the \( y \)-roots, for \( 1 \leq N \leq l \).

   For \( k > m/2 \), we solve the factorization problem of the list decoding by the following algorithm.
Algorithm 3’

**Input:** The polynomial $Q_0(x)$ such that $\deg Q_0 = m$.

**Parameters:** integer $k$ with $k > m/2$.

1. **Step $F_1$**
   - Factorize the univariate polynomials $Q_0(x)$, list all the pairs $\{f, h\}$ of monic factors such that $Q_0(x) = f(x).h(x)$, with $\deg f \leq \deg h$ and $\deg f \leq m/2$.
   - Define $C_1 = \{(f(x), h(x)) | 1 \leq \deg f \leq \deg h < k\}$ and $C_2 = \{(f(x), h(x)) | 1 \leq \deg f < k \leq \deg h\}$

2. **Step $F_2$**
   - $i = 1, \quad i' = 0$
   - (a) For $\{f, h\} \in C_1$
     - If $(y + f(x)) | Q(x, y)$
       - $Z_i = -f(x)$
     - If $i < l$
       - $i = i + 1$
     - If $(y + h(x)) | Q(x, y)$
       - $Z_i = -h(x)$
     - If $i < l$
       - $i = i + 1$

   - $i' = i$

   - (b) For $\{f, h\} \in C_2$
     - If $(y + f(x)) | Q(x, y)$
       - $Z_i' = -f(x)$
     - If $i' < l$
\[ i' = i' + 1 \]

Otherwise

**Output:** \( Z_1, \ldots, Z_N \) the \( y \)-roots for \( 1 \leq N \leq l \).

We denote by \( M(n) \) the minimal number of arithmetic operations necessary to multiply two polynomials of degree less than \( n \) over \( \mathbb{F}_q \), this can be done in \( \mathcal{O}(n \log n \log \log n) \) field operations by the so-called Shonhage-Strassen method and that from [7] Theorem 2.13.

### 5.3 The Cost of the Algorithm

**Theorem 44** The average number of \( \mathbb{F}_q \)-field operations used by the Algorithm 3 or the Algorithm 3’ is

\[ \mathcal{O}(mm'/2 + m^2 \log q) \text{ operations}, \]

where \( m \) is the degree of the first nonzero polynomial \( Q_0(x) \) of \( Q(x, y) \), \( m' = \text{deg } Q(x, y) \), and \( m'' = \mathcal{O}(lM(m') \log m' + m'M(l) \log l) \).

**Proof.** The cost of the step F1 is the cost of the factorization of the univariate polynomial \( Q_0(x) \) of degree \( m \) over \( \mathbb{F}_q \) it can be done by the algorithm of Kaltofen-Shoup [21] using \( \mathcal{O}(m^{1.815} \log q) \) operations. The cost of the step F2 is the cost of the test \( (y + f(x))|Q(x, y) \) which we multiply by the average number of pairs. The polynomial \( Q(x, y) \) is of degree in \( x \) \( m' \) and in \( y \) at most \( l \), since \( f|Q_0 \) then \( \deg f < m' \) and then the test \( y + f(x)|Q(x, y) \) can be done in \( \mathcal{O}(lM(m') \log m' + m'M(l) \log l) \) operations using the modular gcd algorithm (corollary 11.9) of Von Zur Gathen and Gerhard [44]. We have \( m' \geq l \) and that from equation 4.3, furthermore the average number of pairs is equal to \( m/2 \) and the proof is given in the next lemma. Then in total the
algorithm 3 or 3’ can be done with an expected number of operations

\[ \mathcal{O}(mm''/2 + m^2 \log q). \]

The proof of the following lemma is similar to the lemma 5.1. given by Gao and Lauder in [10] in order to analyze the cost of the Hensel lifting. The only difference is that they prove the result for the squarefree univariate polynomial, we remarked that the proof can be extended to the non-squarefree polynomials.

**Lemma 45** The average number of unordered, non-trivial pairs of monic factors \( \{f, h\} \) of a monic polynomial \( g \in \mathbb{F}_q[x] \) of degree \( m \geq 1 \) over the field \( \mathbb{F}_q \) is \( \lfloor m/2 \rfloor \).

**Proof.** We denote by \( P(m, q) \) the set of all monic polynomials of degree \( m \) over \( \mathbb{F}_q \). It is well known that \( |P(m, q)| = q^m \). Let \( g \in P(m, q) \), then we need to find the average number of monic factors of \( g \) whose degree is at least 1, but not greater than \( \lfloor m/2 \rfloor \). This is \( |P(m, q)| = q^m \) divided into the following expression.

\[
\sum_{g \in P(m, q)} \sum_{i=1}^{\lfloor m/2 \rfloor} \sum_{f \in P(i, q), f/g} 1 = \sum_{i=1}^{\lfloor m/2 \rfloor} \sum_{f \in P(i, q)} \sum_{h \in P(m-i, q)} 1 = \sum_{i=1}^{\lfloor m/2 \rfloor} q^i q^{m-i} = q^m \lfloor \frac{m}{2} \rfloor.
\]

Hence the result. 

By applying the Lemma 45 to \( g = Q_0 \), (the first nonzero coefficient of \( Q(x, y) \) under the assumption that \( Q_0 \) is monic ), this gives the result of Theorem 44.
Remark 46  We compare the result of Theorem 44 to the complexity of \cite{31}, $O(n^2 l^2 \log l^2 \log q)$ given by Roth and Ruckenstein\textsuperscript{1}, and which works for the small rates, and then from the condition 2 in Theorem 33 we have $m' = \deg Q_x(x, y) < n - \tau$, and $m = \deg Q_0(x) < n - \tau$. With $n - \tau < \frac{n + k}{2}$ and then since $n < q$ running time of the algorithm 3 or 3’ in average is

$$O(n l M(n) \log n + n^2 M(l) \log l + n^2 \log q) = O(n M(n) M(l) \log q).$$

\textsuperscript{1}The algorithm of Roth Ruckenstein was improved by McEliece in \cite{26} to work for high rate, and proposed as a solution for the factorization step in the Guruswami–Sudan Algorithm, the complexity of the improved algorithm is not mentioned.
Chapter 6

Goppa Codes: Construction and Properties

6.1 Some Notions from Algebraic Geometry

In the following section we give a brief introduction to some notions from algebraic geometry, in order to construct the so-called algebraic geometric codes (AG codes), also called geometric Goppa codes or simply Goppa codes. More detailed results and lectures are given in [36] and [37].

Definition 47 Let $\overline{\mathbb{F}}_q$ be the algebraic closure of $\mathbb{F}_q$. The $n$ dimensional affine space $\mathbb{A}^n$ over $\overline{\mathbb{F}}_q$ is the set

$$\mathbb{A}^n = \{(a_1, \ldots, a_n) \mid a_i \in \overline{\mathbb{F}}_q\}.$$  

The elements of $\mathbb{A}^n$ are called affine points. And if the coordinates of a point $P \in \mathbb{A}^n$ are all in $\mathbb{F}_q$, then the point $P$ is called $\mathbb{F}_q$-rational.
On the set $A^{n+1} \setminus \{(0, \ldots, 0)\}$, an equivalence relation is given by:

$$(a_0, \ldots, a_n) \equiv (b_0, \ldots, b_n) \iff \exists \lambda \neq 0 \in \mathbb{F}_q \text{ such that } b_i = \lambda a_i \text{ for } 0 \leq i \leq n.$$ 

The set of all equivalence classes $\mathbb{P}^n := A^{n+1} \setminus \{(0, \ldots, 0)\}/\equiv$ is called the **projective space** of dimension $n$.

An element $P \in \mathbb{P}^n$ is called a **projective point** or simply point, and it is denoted by

$$P = (a_0 : \ldots : a_n).$$

The point $P = (a_0 : \ldots : a_n) \in \mathbb{P}^n$ is said to be $\mathbb{F}_q$-**rational** if there exist an $(n + 1)$-uple of homogeneous coordinates $(\lambda a_0, \ldots, \lambda a_n)$ with $\lambda \neq 0$ and $\lambda a_i \in \mathbb{F}_q$ for all $i \in \{0, \ldots, n\}$. This is equivalent to saying that if $a_i \neq 0$, then $a_j/a_i \in \mathbb{F}_q$ for all $j \in \{0, \ldots, n\}$.

Let $d \in \mathbb{N}$ and $G$ be a homogeneous polynomial in $\mathbb{F}_q[x_0, \ldots, x_n]$ of degree $d$, we call $P = (a_0 : \ldots : a_n) \in \mathbb{P}^n$ a **zero** of $G$, and we write

$$G(P) = 0 \quad \text{if} \quad G(a_0 : \ldots : a_n) = 0.$$ 

This make sense, since $G(\lambda a_0, \ldots, \lambda a_n) = \lambda^d G(a_0, \ldots, a_n)$.

**Definition 48** *A subset $V$ in $\mathbb{P}^n$ is an algebraic set*, if there exists a set $T$ of homogeneous polynomials in $\mathbb{F}_q[x_0, \ldots, x_n]$, such that

$$V = \{P \in \mathbb{P}^n \mid G(P) = 0, \forall G \in T\}.$$ 

**Definition 49** *Given an algebraic set $V \subset \mathbb{P}^n$, the set of polynomials

$$I(V) = \{G \in \mathbb{F}_q[x_0, \ldots, x_n] \mid G(P) = 0, \forall P \in V\}$$

is called the **ideal** of $V$.**
$I(V)$ is an ideal in $\mathbb{F}_q[x_0, \ldots, x_n]$, which is a Noetherian ring by the Hilbert basis theorem, so can be generated by a finite set of polynomials $F_1, \ldots, F_r$ and

$$V = \{ P \in \mathbb{P}^n \mid F_1(P) = \ldots = F_r(P) = 0 \}.$$

An algebraic set $V \subset \mathbb{P}^n$ is called a projective variety if the corresponding ideal $I(V)$ is a prime ideal. A projective variety $V$ is said to be defined over $\mathbb{F}_q$ if the generators of $I(V)$ are in $\mathbb{F}_q[x_0, \ldots, x_n]$, and then we denote $I(V)$ by $I_{\mathbb{F}_q}(V)$. In this case

$$I_{\mathbb{F}_q}(V) = I(V) \cap \mathbb{F}_q[x_0, \ldots, x_n].$$

The subset of $\mathbb{F}_q$-rational points of the variety $V$ is denoted $V(\mathbb{F}_q)$.

The action of $\text{Gal}(\mathbb{F}_q/\mathbb{F}_q)$ extends naturally to the action on the set $\mathbb{P}^n(\mathbb{F}_q)$ by the following way:

For $P \in \mathbb{P}^n$ and $\sigma \in \text{Gal}(\mathbb{F}_q/\mathbb{F}_q)$, then

$$\sigma(P) = (\sigma(a_0) : \ldots : \sigma(a_n)).$$

And then we can deduce that the subset of the $\mathbb{F}_q$-rational points of a projective variety $V$ is

$$V(\mathbb{F}_q) = \{ P \in V \mid \sigma(P) = P, \forall \sigma \in \text{Gal}(\mathbb{F}_q/\mathbb{F}_q) \}.$$

**Definition 50** Let $V$ be a non-empty variety in $\mathbb{P}^n$. The homogeneous coordinate ring is defined to be $\mathbb{F}_q[V] = \mathbb{F}_q[x_0, \ldots, x_n]/I(V)$. If $V$ is defined over $\mathbb{F}_q$, then $\mathbb{F}_q[V] = \mathbb{F}_q[x_0, \ldots, x_n]/I_{\mathbb{F}_q}(V)$.

---

1 Some texts define this as an irreducible variety, where the variety is defined as an algebraic set.
An element \( f \in \overline{\mathbb{F}}_q[V] \) is said to be a form of degree \( d \), if \( f = F + I(V) \) for some homogeneous polynomial \( F \in \mathbb{F}_q[x_0, \ldots, x_n] \), with \( \deg F = d \).

\[ f(P) \neq 0 \] signifies that if \( P \in V \) then \( F(P) \neq 0 \).

The function field of the variety \( V \) is defined by

\[ \mathbb{F}_q(V) = \{ g/h | g, h \in \overline{\mathbb{F}}_q[V] \text{ are forms of the same degree and } h \neq 0 \}. \]

It is a subfield of the quotient field of \( \overline{\mathbb{F}}_q[V] \). The dimension of \( V \) is the transcendence degree of \( \mathbb{F}_q(V) \) over \( \mathbb{F}_q \).

We say that \( f \in \mathbb{F}_q(V) \) is defined at \( P \in V \) and we call \( f(P) \) the value of \( f \) at \( P \), if there exists two forms \( g, h \) with the same degree such that \( f = g/h \), \( h(P) \neq 0 \) and \( g = G + I(V) \), \( h = H + I(V) \) and \( f(P) = (G/H)(P) \).

A very interesting special case of projective varieties are the projective curves \( C \) on \( \mathbb{P}^n \), which are the projective varieties of dimension 1. A plane projective curve is a projective variety \( C \) on \( \mathbb{P}^2 \). It is the set of zeroes of an irreducible homogeneous polynomial \( C \in \mathbb{F}_q[x_0, x_1, x_2] \), if \( C \) is defined over \( \overline{\mathbb{F}}_q \) or the set of the zeroes of an irreducible homogeneous polynomial \( C \in \mathbb{F}_q[x_0, x_1, x_2] \) if \( C \) is defined over \( \mathbb{F}_q \).

**Example 51** Let \( C \) be a curve defined by the polynomial

\[ F(x, y, z) = y^2z - yz^2 + x^3 - x^2z \] over the field \( \overline{\mathbb{F}}_2 \).

Consider the rational function \( f \) represented by \( (y^2 + yz)/zx \). \( f \) is defined at the point \( P = (0 : 0 : 1) \in C \) because \( f \) is represented by \( (x^2 - xz)/x^2 \) as well. In fact, \( x^2(y^2 + yz) - zx(x^2 - xz) \in I \) as \( z^2(y^2 + yz - zx)(x^2 - xz) = zF \). Therefore \( f \) is defined at \( P \) and \( f(P) = 0 \).

**Remark 52** The function field \( \mathbb{F}_q(C) \), respectively \( \mathbb{F}_q(C) \) can be seen as a
finite algebraic extension of $\overline{\mathbb{F}}_q(x)$ respectively $\mathbb{F}_q(x)$ for some $x$ in $\overline{\mathbb{F}}_q(C)$ respectively in $\mathbb{F}_q(C)$ which is transcendental over $\mathbb{F}_q$. Now because $I(C)$ respectively $I_{\mathbb{F}_q}(C)$ is finitely generated, this gives us the following representation of the function fields

$$\overline{\mathbb{F}}_q(C) = \overline{\mathbb{F}}_q(x)[y_1, \ldots, y_n].$$

$$\mathbb{F}_q(C) = \mathbb{F}_q(x)[y_1, \ldots, y_n],$$

(6.1)

where each $y_i$ is algebraic over $\overline{\mathbb{F}}_q(x)[y_1, \ldots, y_{i-1}]$, respectively over $\mathbb{F}_q(x)[y_1, \ldots, y_{i-1}]$ i.e., each $y_i$ satisfies a polynomial equation over $\overline{\mathbb{F}}_q(x)[y_1, \ldots, y_{i-1}]$.

**Definition 53** Let $V$ be a projective variety, and $P \in V$. We define the **local ring** at the point $P$ by

$$\mathcal{O}_P(V) = \{f/g \mid f, g \in \mathbb{F}_q[V] \text{ are forms of the same degree and } g(P) \neq 0\}.$$ 

**Remark 54** $\mathcal{O}_P(V)$ is a local ring, and its unique maximal ideal is

$$M_P(V) = \{f/g \in \mathcal{O}_P(V) \mid f(P) = 0 \text{ and } g(P) \neq 0\}.$$ 

**Definition 55** Let $V \subset \mathbb{P}^n$ be a variety, $\{G_1, \ldots, G_s\}$ a set of generators of the ideal $I(V)$, and $P \in V$. We consider the matrix $\mathcal{J}_{V,P} = (a_{i,j})$ where $a_{i,j} = (\partial G_i/\partial x_j)(P)$, $i \in [s]$ and $0 \leq j \leq n$. The point $P$ is called a **non-singular or simple point** if

$$\text{rank } (\mathcal{J}_{V,P}) = n - \text{dim } V.$$ 

If not, the point $P$ is called a **singular point**. The variety $V$ is **singular** if it has at least a singular point; if not it is called **smooth**. The matrix $\mathcal{J}_{V,P}$ is called the **Jacobian matrix** at the point $P$. The rank of $\mathcal{J}_{V,P}$ is independent of the choice of the set of generators of the ideal $I(V).$
Example 56 Let $\mathcal{C} = \{ P \in \mathbb{P}^2 \mid C(P) = 0 \}$ be a projective plane curve that is a projective variety of dimension 1, then the Jacobian matrix at $P = (a : b : c)$ is

$$J_{V,P} = \begin{pmatrix} C_{x_0}(a,b,c) \\ C_{x_1}(a,b,c) \\ C_{x_2}(a,b,c) \end{pmatrix}. $$

Where $C_{x_i}$ for $i \in \{1, 2, 3\}$ are the usual derivatives. Then,

$P = (a : b : c) \in \mathcal{C}$ is simple $\iff C_{x_i}(a,b,c) \neq 0$ for at least one $i \in \{1, 2, 3\}$.

Following the previous equivalence the curve $\mathcal{C} = \{ (x_0 : x_1 : x_2) \in \mathbb{P}^2 \mid x_0^m + x_1^m + x_2^m = 0 \}$ is smooth over the field of characteristic $p$, except when the gcd $(m, p) \neq 0$; in that case every point of the curve is singular.

### 6.1.1 Characterisation of the Singular Points of an Algebraic Curve

**Definition 57** Let $A$ be a Noetherian local ring with maximal ideal $\mathcal{M}$ and residue field $K = A/\mathcal{M}$. $A$ is called a **regular ideal ring** if the dimension of the $K$-vector space $\mathcal{M}/\mathcal{M}^2$ is the same as $\dim A$.

The next theorem shows that the non-singular points of a curve are exactly the points for which their local rings are regular rings.

**Theorem 58** Let $\mathcal{C}$ be a projective curve and $P$ be a point of $\mathcal{C}$. Then $\mathcal{C}$ is non-singular at $P$ if and only if the local ring $\mathcal{O}_P(\mathcal{C})$ of $P$ is a regular ring, i.e.,

$P$ is non singular $\iff \mathcal{O}_P(\mathcal{C})$ is a regular ring.

**Proof.** Hartshorne [[17] Chap.1, Th.5.1]
Definition 59  A valuation ring of the function field $\mathbb{F}_q(C)$ is a ring $\mathcal{O}$ with the properties

1. $\mathbb{F}_q \subset \mathcal{O} \subset \mathbb{F}_q(C)$.

2. For any $z \in \mathbb{F}_q(C)$, $z \in \mathcal{O}$ or $z^{-1} \in \mathcal{O}$.

Theorem 60  ([37]Th 1.1.6) Let $\mathcal{O}$ be a valuation ring of the function field $\mathbb{F}_q(C)$. Then

1. $\mathcal{O}$ is a local ring and has a unique maximal ideal $\mathcal{M} = \mathcal{O} \setminus \mathcal{O}^*$, where $\mathcal{O}^* = \{z \in \mathcal{O} | \exists \omega \in \mathcal{O} : z\omega = 1\}$

2. For $0 \neq x \in \mathbb{F}_q(\mathcal{O})$, $x \in \mathcal{M} \iff x^{-1} \notin \mathcal{O}$

3. $\mathcal{M}$ is a principal ideal

4. If $\mathcal{M} = t\mathcal{O}$ then any $0 \neq z \in \mathbb{F}_q(C)$ has a unique representation of the form $z = t^n u$ for some $u \in \mathbb{Z}, u \in \mathcal{O}^*$

5. $\mathcal{O}$ is a principal ideal domain. If $\mathcal{M} = t\mathcal{O}$ and $\{0\} \neq \mathcal{I} \subset \mathcal{O}$ is an ideal then $\mathcal{I} = t^n \mathcal{O}$ for some $n \in \mathbb{N}$.

The representation of the elements of $\mathbb{F}_q(C)$ given by the previous theorem, defines a function

$$v : \mathbb{F}_q(C) \longrightarrow \mathbb{Z} \cup \{\infty\},$$

such that if $z = t^n u$ with $u \in \mathcal{O}^*, n \in \mathbb{Z}$, then $v(z) = n$ and $v(0) = \infty$. This function $v$ is called discrete valuation. It is surjective with the properties given by the following theorem.

Theorem 61  ([37]ThI.1.12) The discrete valuation $v : \mathbb{F}_q(C) \longrightarrow \mathbb{Z} \cup \{\infty\}$ satisfies

- $v(x) = \infty \iff x = 0$
- $v(x.y) = v(x) + v(y)$ for any $x, y \in \mathbb{F}_q(C)$
The most remarkable characteristic of a non-singular point of a curve is the following theorem.

**Theorem 62** [36] [Th.4.9]

Let $C$ be an algebraic curve. A point $P$ of $C$ is non-singular if and only if the local ring $\mathcal{O}_P(C)$ is a discrete valuation ring.

The proof of this theorem is essentially based on a result of Atiyah-Macdonald given in [1][Pro 9.2 p94], where a Noetherian local ring, is characterized to be a discrete valuation ring if and only if it is a regular local ring. Theorem 62 then follows from the Theorem 58.

A point $P$ of $C$ is said to be a **zero of multiplicity** $m$ of $f \in \mathbb{F}_q(C)$ if $\nu_P(f) = m > 0$; a **pole of multiplicity** $-m$ of $f$ if $\nu_P(f) = m < 0$. More generally the **order** of $f$ at the point $P \in C$ is $\nu_P(f)$.

Often $\mathcal{M}_P(C)$ is called the place $P$.

**Remark 63** If we consider a smooth projective curve defined over $\mathbb{F}_q$ we can see it is a curve over $\mathbb{F}_q$ of which we can see only a fraction of all points. From theorem 62 we have over $\mathbb{F}_q$ a one-to-one correspondence between the points of $C$ and the discrete valuation ring of the function field. Since we cannot see all points, this is not true over $\mathbb{F}_q$, but nevertheless we can look at all discrete valuation rings contained in $\mathbb{F}_q(C)$, such that the discrete valuation is trivial on $\mathbb{F}_q$.s In this way to each point $P$ of $C$ we can associate its discrete valuation ring $\mathcal{O}_P$ with maximal ideal $\mathcal{M}_P$. The field $\mathbb{F}_P = \mathcal{O}_P/\mathcal{M}_P$ is called the **residue class field.** This is a finite extension of $\mathbb{F}_q$, and $d = [\mathcal{O}_v/\mathcal{M}_v : \mathbb{F}_q]$ the **degree of the point P**. Now the residue class field associated to a
point of degree \( d \) can be identified to \( \mathbb{F}_{q^d} \), which is a cyclic extension of \( \mathbb{F}_q \) with group \( \text{Gal}(\mathbb{F}_{q^d}/\mathbb{F}_q) = \{\sigma, \sigma^2, \ldots, \sigma^{d-1}\} \). Then to each point \( P \) of \( \mathbb{C}_{\mathbb{F}_q} \) of degree \( d \) there corresponds a point \( P_0 \) of \( \mathbb{C}_{\mathbb{F}_{q^d}} \) and its \( d-1 \) conjugates \( \{\sigma(P_0), \ldots, \sigma^{d-1}(P_0)\} \), the \( d \) points are of degree one over \( \mathbb{F}_{q^d} \). Then the point of degree one over \( \mathbb{F}_q \) also called place of degree one correspond to the \( \mathbb{F}_q \)-rational points.

6.1.2 Divisors on Algebraic Curves

Let \( \mathcal{C} \) be a smooth projective curve defined over \( \mathbb{F}_q \), a divisor of \( \mathcal{C} \) is a formal sum

\[
D = \sum a_P P,
\]

where the sum is over all points on \( \mathcal{C} \), the coefficients are integers and are zeros for all but a finite number of \( P \)'s.

An \( \mathbb{F}_q \)-rational divisor on \( \mathcal{C} \) is a divisor \( D = \sum a_P P \), such that

\[
D = \sigma(D) \text{ where } \sigma(D) = \sum a_P \sigma(P) \forall \sigma \in \text{Gal}(\mathbb{F}_{q^d}/\mathbb{F}_q).
\]

If \( D = \sum a_P P \) and \( D' = \sum a'_P P \),

\[
D + D' = \sum (a_P + a'_P) P,
\]

and then the set of divisors of \( \mathcal{C} \) is a free abelian group denoted by \( \text{Div}(\mathcal{C}) \).

The degree of \( D \) is

\[
\text{deg } (D) = \sum a_P \text{deg } P.
\]

The set \( \text{supp } D = \{P \in \mathcal{C} \mid a_P \neq 0\} \) is called the support of \( D \). The set of divisors of \( \mathcal{C} \) is a free abelian group denoted by \( \text{Div}(\mathcal{C}) \) for which we can
define a partial ordering by,

\[ D \geq D' \iff D - D' \geq 0, \]

where \( D \geq 0 \) is defined by \( a_P \geq 0 \) for all \( P \) in \( \text{supp} \ (D) \).

Given \( 0 \neq f \in \mathbb{F}_q(C) \), the **principal divisor** is defined to be

\[ (f) = \sum v_P(f) P, \]

where \( v_P(f) \) is defined in theorem 60.

**Remark 64** As a consequence of the residue theorem we have that every function \( f \) in \( \mathbb{F}_q(C) \) has an equal number of zeros and poles, hence \( \deg ((f)) = 0 \).

### 6.1.3 Linear Systems

Let \( G \) be a rational divisor on a curve \( C \). Consider the following vector space over \( \mathbb{F}_q \) of dimension \( l(G) \),

\[ \mathcal{L}(G) = \{ f \in \mathbb{F}_q(C) \mid (f) + G \geq 0 \} \cup \{0\}. \]

A function \( f \in \mathcal{L}(G) \) has the properties that it has poles only at the zeroes of the divisor \( G \) and has zeroes at least at the poles of \( G \). The result of the following lemma will be used to proof the properties of Goppa codes.

**Lemma 65** [37][lemma1.4.7]

If \( G < 0 \), then \( \mathcal{L}(G) = 0 \)

**Proof.** Assume that \( G < 0 \) and let \( f \in \mathcal{L}(G) \) such that \( f \neq 0 \), then \( (f) + G \geq 0 \) i.e., \( \deg ((f) + G) \geq 0 \), but \( \deg ((f) + G) = \deg ((f)) + \deg (G) \geq 0 \). By remark 64 \( \deg ((f)) = 0 \). Absurd. ■
The \textbf{genus} of the curve $C$ is defined by

$$g = \max \{ \deg G - l(G) + 1 \mid G \in \text{Div}(C) \}.$$ 

A canonical divisor $W$ is a divisor on $C$ such that

$$l(W) = g \text{ and } \deg W = 2g - 2.$$ 

(6.2)

The next theorem called the Riemann–Roch theorem allows us to determine $\dim \mathcal{L}(G) = l(G)$, and then to determine the parameters of the Goppa codes.

\textbf{Theorem 66 (Riemann–Roch)} Let $W$ be a canonical divisor of a curve $C$ of genus $g$. For any divisor $G \in \text{Div}(C)$, we have

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

\textbf{Proof.} [37][Th. I.5.15]

Some consequences of the Riemann–Roch theorem are given in the following corollary.

\textbf{Corollary 67} We have

1. $l(W - G) = g - 1 - \deg G$ \hspace{1em} for $\deg G < 0$;
2. $l(G) = \deg G - g + 1$ \hspace{1em} for $\deg G > 2g - 2$;
3. $l(G) = g - 1$ \hspace{1em} if $G \neq W$ and $\deg G = 2g - 2$;

\textbf{Proof.} [37] [Th.1.5.17 and Pro.1.6.2]
6.2 Construction of the Goppa Codes

For the construction of the Goppa codes we will be interested only in the smooth projective curve over $\mathbb{F}_q$, that is a smooth projective variety of dimension 1 over $\mathbb{F}_q$. The Goppa codes considered here are the Goppa codes obtained from the so-called $\mathcal{L}$-construction, there is another class of Goppa codes dual to this class, obtained by computing residues of differential forms on $\mathcal{C}$. For more details see [36], [42] and [37]. However this is not an essential difference since such codes can be defined in either way.

Let $\mathcal{C}$ be a smooth projective curve, defined over a finite field $\mathbb{F}_q$. Let $P_1, \ldots, P_n$ be $\mathbb{F}_q$-rational points of $\mathcal{C}$, and set $D = P_1 + \ldots + P_n$.

Let $G$ be an $\mathbb{F}_q$-rational divisor of the curve $\mathcal{C}$ such that $\deg G \geq 0$. We assume that $G$ has support disjoint from the divisor $D$, i.e.,

$$\text{supp} D \cap \text{supp} G = \emptyset.$$ 

**Definition 68** The linear code $C_{\mathcal{L}}(D, G)$ is the image of the linear map,

$$\text{ev} : \mathcal{L}(G) \longrightarrow \mathbb{F}_q^n$$

$$f \mapsto (f(P_1), \ldots, f(P_n)).$$ (6.3)

Such a code is called a **Geometric Goppa code** or simply Goppa code.

The definition make sense since the support of $G$ and $D$ are distinct, which implies that the support of $G$ and $P_i$ are also distinct. Because $f \in \mathcal{L}(G)$, then $v_{P_i}(f) \geq 0$ and then $f$ has no poles at $P_i$ i.e., $f$ is defined at $P_i$, by hypothesis we also assumed that $\deg P_i = 1$, and then $f(P_i)$ is in $\mathbb{F}_q$, (as the coefficient of the rational function and the coordinates of $P_i$ are in $\mathbb{F}_q$).

---

\(^2\)In [27] Miura has shown that the construction of Goppa codes can be done over singular curves, however in [25] Matsumoto proved that the codes constructed on smooth curve have better parameters.
The properties of the Goppa codes are given by the following theorem.

**Theorem 69** \cite{37}[Th II.2.2] Let \( C \) be a smooth projective curve of genus \( g \), and \( P_1, \ldots, P_n \) be \( \mathbb{F}_q \)-rational points of \( C \). Then \( C_L(D,G) \) is a linear \([n, k, d]_q\)-code with

\[
k = l(G) - l(G - D) = \deg G - g + 1 + l(W - G) - l(G - D)
\]

and

\[
d \geq n - \deg G.
\]

Suppose \( \deg G < n \). The evaluation map

\[
ev : L(G) \longrightarrow \mathbb{F}_q^n
\]

is then an embedding, and we have:

1. \( C_L(D, G) \) is an \([n, k, d]_q\)-code with \( k \geq \deg G - g + 1 \) and \( d \geq n - \deg G \),

2. if in addition \( 2g - 2 < \deg G < n \), then

\[
k = \deg G - g + 1
\]

3. if \( \{f_1, \ldots, f_k\} \) is a basis of \( L(G) \), then the matrix

\[
\begin{pmatrix}
  f_1(P_1) & \cdots & f_1(P_n) \\
  \vdots & \ddots & \vdots \\
  f_k(P_1) & \cdots & f_k(P_n)
\end{pmatrix}
\]

is a generator matrix of \( C_L(D, G) \).

**Proof.** Since \( C_L(D, G) \) is the direct image of the evaluation map then we have a surjective linear map from \( L(G) \) to \( C_L(D, G) \) defined by the restriction
of the evaluation map, and then $C_L(D,G)$ is linear. The kernel is

$$\ker(ev) = \{ f \in \mathcal{L}(G) \mid \nu_{P_i}(f) > 0 \text{ for } 1 \leq i \leq n \} = \mathcal{L}(G - D). \quad (6.4)$$

Observe that because $f \in \ker(ev)$ by definition $f \in \mathcal{L}(G)$ and $f(P_i) = 0$ for all $i \in [n]$, so that $(f) \geq -G$ and the $P_i$’s are zeros of $f$ and then $\nu_{P_i}(f) > 0$ i.e., the coefficients of $P_i$ in the principal divisor $(f)$ is at least 1. But $P_i \notin \text{supp } G$, and therefore $(f) + G - P_1 - P_2 - \ldots - P_n \geq 0$, which is equivalent to say that $f \in \mathcal{L}(G - D)$.

From the discussion above, we deduce that

$$C_L(D,G) \cong \mathcal{L}(G)/\mathcal{L}(G - D) \quad (6.5)$$

and hence,

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

By the Riemann–Roch theorem

$$k = \deg G - g + 1 + \mu(W - G) - \mu(G - D). \quad (6.6)$$

This proves the assertion on the dimension of $C_L(D,G)$.

Let $f \in \mathcal{L}(G)$, be such that weight of $ev(f)$ is $d$. Then, there are exactly $n - d$ places $P_{i_1}, \ldots, P_{i_{n-d}}$ such that $f(P_{i_j}) = 0$ for $j \in [n-d]$. By the same argument as for the $\ker(ev)$, we have that

$$f \in \mathcal{L}
\left(G - \sum_{\tau=1}^{n-d} P_{i_{\tau}}\right).$$
By taking degrees (and since $f \neq 0$ by the lemma 65 we must have

$$\deg G - n + d \geq 0,$$

so we will have $f \equiv 0$, and hence $d \geq n - \deg G$.

Now if we suppose that the degree of $G$ is strictly less than $n$, we show that the evaluation map is an injective map. We have already shown that $\ker(ev) = \mathcal{L}(G - D)$. Let $f$ be in $\mathcal{L}(G - D)$, then by the same argument as for lemma 65 if we assume that $\deg G < n$ we will get that $f \equiv 0$, which shows that the evaluation map in this case is injective, and then from the equality 6.6 on the dimension, we get

$$k = \dim C_L(D, G) = \dim \mathcal{L}(G) \geq \deg G + 1 - g.$$

If we assume now that $\deg G > 2g - 2$, then from Corollary 67 we have $l(G) = \deg G + 1 - g$. We obtain $k = l(G) = \deg G + 1 - g = \dim C_L(D, G)$.

For the generator matrix, since we are in the case that

$$2g - 2 < \deg G < n.$$

From the previous results on the $\ker(ev) = \{0\}$ we have that the equivalence 6.5 becomes $C_L(D, G) \cong \mathcal{L}(G)$. Then if $\{f_1, \ldots, f_k\}$ is a basis of $\mathcal{L}(G)$, the vectors $(f_1(P_1), \ldots, f_1(P_n)), \ldots, (f_k(P_1), \ldots, f_k(P_n))$ form a basis for the linear code $C_L(D, G)$, which gives the result on the generator matrix, from the chapter 2.

The integer $d^* = n - \deg G$ is called the designed distance of the code $C_L(D, G)$. The theorem states that the minimum distance $d$ of a geometrical Goppa code cannot be less than its designed distance. The question whether $d^* = d$ or $d^* < d$ is answered by the following lemma.
Lemma 70 ([37] Th II.2.5) Suppose that \( \dim G > 0 \) and \( d^* = n - \deg G > 0 \). Then \( d = d^* \) if and only if there exists a divisor \( D' \) with \( 0 \leq D' \leq D \), \( \deg D' = \deg G \) and \( \dim \mathcal{L}(G - D') > 0 \).

Proof.

Assume that \( d = d^* \) then there exists a codeword in \( C_L(D, G) \) with weight \( d \) i.e., there exists \( 0 \neq f \in \mathcal{L}(G) \) such that \( f(P_i) \neq 0 \) for exactly \( d \) values of \( i \in [n] \) or equivalently \( n - d \) components of the codeword \( (f(P_1), \ldots, f(P_n)) \) are equal to zero. As \( n - d = \deg G \), then we have \( f(P_{i_j}) = 0 \) for \( j = 1, \ldots, \deg G \) components. We set \( D' = \sum_{i=1}^{\deg G} P_{i_j} \). Then \( 0 \leq D' \leq P_1 + \ldots + P_n = D \), \( \deg D' = \deg G \) and \( \dim (G - D') > 0 \) (as \( f \in \mathcal{L}(G - D') \)).

Conversely, if \( D' \) has the above properties, then \( \mathcal{L}(G - D') \neq \{0\} \), since \( \dim (G - D') > 0 \). Hence there exists \( 0 \neq f \in \mathcal{L}(G - D') \), and the weight of the corresponding codeword \( (f(P_1), \ldots, f(P_n)) \) is less or equal than \( n - \deg D' \) but \( \deg G = \deg D' \) i.e., we have \( d \leq d^* \) since from theorem 69 we have \( d \geq n - \deg G = d^* \), then equality follows. ■

The \([n, k]\) RS codes can be described as a special case of Goppa codes. The corresponding curve is the projective line, which is of genus 0, and the function field \( \mathbb{F}_q(C) \) is \( \mathbb{F}_q(x) \). All places of degree 1 of \( \mathbb{F}_q(x) \) are the \( P_i = x - p_i \) with \( p_i \in \mathbb{F}_q \), plus the place at infinity \( P_\infty \) the place associated with \( 1/x \). The divisor \( G \) is \((k - 1)P_\infty \), and \( \mathcal{L}(G) = \mathcal{L}_k \) is the set of polynomials of degree less than \( k \) over \( \mathbb{F}_q \).
6.3 The Goppa Code Family is a Good Family

6.3.1 Number of Rational Points

Remark 71 From the construction of the Goppa code $C_L(D,G)$, we can deduce that the length $n$ of these codes is bounded by the number of places of degree one. Hence the necessity to construct codes over curves with many $\mathbb{F}_q$ rational points.

In this paragraph we will give a brief survey on the number of rational points on a curve.

F.K. Schmidt\(^3\) introduced the zeta function for a smooth projective curve $C$ over $\mathbb{F}_q$ defined by

$$Z_C(t) = \exp\left(\sum_{r=1}^{\infty} N_{qr^r} \frac{t^r}{r}\right),$$

where $N_{qr^r}$ is the number of $\mathbb{F}_{qr^r}$-rational point over $C$. He observed that the theorem of Riemann–Roch implied that for a curve of genus $g$, this function has the form

$$Z_C(t) = \frac{L_C(t)}{(1 - t)(1 - qt)}, \quad (6.7)$$

where $L_C(t)$ is a polynomial of degree $2g$. Furthermore the function zeta satisfies the functional equation

$$Z_C(1/qt) = q^{1-g}t^{2-2g}Z_C(t).$$

Later A. Weil proved that the polynomial in the equation (6.7) is a polynomial

\(^3\)Detailed definitions results and references are given in [37].
with integral coefficients of the form

\[ L_C(t) = \prod_{i=1}^{2g} (1 - \alpha_i t), \]

where the \( \alpha_i \) are algebraic integers with \( |\alpha_i| = \sqrt{q} \). This implies a bound on the number of places of degree one, called the \textbf{Hasse-Weil-bound}.

**Theorem 72 (Hasse-Weil-bound)**

The number \( N_q \) of places of degree one of the curve \( C \) can be estimated by

\[ |N_q - (q + 1)| \leq 2g\sqrt{q}. \]

**Remark 73** If we apply the Hasse-Weil theorem to find the number \( N_{q^r} \) of \( \mathbb{F}_{q^r} \) rationals points over \( C \) we get

\[ |N_{q^r} - (q^r + 1)| \leq 2g\sqrt{q^r}. \] (6.8)

From remark 63 we have that the points of degree \( r \) of \( C \) over \( \mathbb{F}_q \) corresponds to the points of degree one over \( \mathbb{F}_{q^r} \) and then the estimation given by the inequality 6.8 give us in fact an estimation on the number of points of degree \( r \) on the curve \( C \) over \( \mathbb{F}_q \).

An improvement of the bound was given by \textbf{Serre}, that is

\[ |N_q - (q + 1)| \leq g\lfloor 2\sqrt{q}\rfloor \] (6.9)

The curves of genus \( g \) for which we have \( N_q = q + 1 + 2gq^{1/2} \) are called \textbf{maximal}.

There is a restriction on the genus of the maximal curves, that is \( g \) is bounded
by the following bound
\[ g \leq \frac{q - \sqrt{q}}{2}, \quad (6.10) \]

Detailed proofs of the equations 6.9 and 6.10 are given in [36].

### 6.3.2 Asymptotic Bounds

From Theorem 69 we have that if \( \deg G < n \), then we have

\[ k \geq \deg G - g + 1, \text{ and } d \geq n - \deg G, \]

and then by summing the two inequalities we get

\[ k \geq n - d - g - 1. \quad (6.11) \]

And so,

\[ R \geq 1 - \delta - \frac{g}{n}. \]

It is clear that \( R \) is maximized if the quantity \( \frac{g-1}{n} \) is minimized i.e., if \( n \) is the largest possible, since \( n \) is bounded by the number of rational points, we have to consider a family of curves with \( N_q = N_q(C) \) as large as possible. We define

\[ A(q) = \limsup_{g \to \infty} \frac{N_q(C)}{g}, \]

where \( C \) runs over all smooth projective curves over \( \mathbb{F}_q \). The real value of \( A(q) \) is unknown, but from the Serre bound we get that

\[ A(q) \leq \lfloor 2\sqrt{q} \rfloor. \]

A mosaic of works on the value of \( A(q) \) was given for special cases of \( q \). In [12] Garcia and Stichtenoth gave a proof that \( A(q) > 0 \) if \( q \) is not a prime
number.

A more interesting bound was given by Drinfeld and Vladut.

**Theorem 74 Drinfeld-Vladut Bound**

For a square $q$ we have

$$A(q) \leq \sqrt{q} - 1.$$  

We have seen that the Goppa construction gives the following inequality for the parameters of an $[n,k,d]$ code $C$:

$$k \geq n - d - (g - 1).$$

By applying this construction to modular curves, Tsfasman-Vladut-Zink construct a sequence of modular curves over $\mathbb{F}_{q^2}$ and proved that in fact:

$$A(q) = \sqrt{q} - 1. \quad (6.12)$$

At the end of the first chapter we were interested in finding a lower bound for $\alpha_q(\delta)$ better than Gilbert Varshamov-bound. The next proposition gives this bound.

**Proposition 75 (corollary 3.4.2 [42])**

Suppose that $A(q) > 1$ then

$$\alpha_q(\delta) \geq 1 - \delta - A(q)^{-1}$$

in that interval $0 \leq \delta \leq 1 - A(q)^{-1}$.

**Proof.** By the remarks above there exists a sequence of smooth projective curves $C_i$ over $\mathbb{F}_q$ of genus $g_i$ with $n_i$ and $g_i$ tending to infinity and $\lim_{i \to \infty} \frac{n_i + 1}{g_i} = A(q)$. Choose an $\mathbb{F}_q$-rational point $Q_i$ on $C_i$ and let $P_i$ be the $n_i$ remaining $\mathbb{F}_q$ rational points on $C_i$. Let $r_i$ be an integer satisfying $2g_i - 2 < r_i < n_i$ then
by theorem (69) the code $C_L(C_i, P_i, r_i Q_i)$ has length $n_i$ and rate

$$R_i = \deg \left( r_i Q_i + 1 - g_i \right)/n_i = (r_i + 1 - g_i)/n_i$$

and relative distance

$$\delta_i \geq 1 - \deg(r_i Q_i)/n_i = 1 - r_i/n_i.$$ 

Thus $R_i + \delta_i \geq \frac{r_i + 1 - g_i}{n_i} + 1 - \frac{r_i}{n_i} = 1 + \frac{1}{n_i} - \frac{g_i}{n_i}$ implying

$$R \geq -\delta + 1 - \frac{1}{A(q)},$$

where $\lim_{i \to \infty} R_i = R$ and $\lim_{i \to \infty} \delta_i = \delta$. Hence, $\alpha_q(\delta) \geq -\delta + 1 - \frac{1}{A(q)}$. 

By using the result (6.12) in the previous proposition we have the following bound called the Tsfasman-Vladut-Zink-Bound, which gives high importance to the Goppa codes.

**Theorem 76** There exists a family of Goppa codes over $\mathbb{F}_q$, with $q = p^2$, such that

$$\alpha_q(\delta) \geq \left(1 - \frac{1}{\sqrt{q} - 1}\right) - \delta$$

for $0 \leq \delta \leq 1 - \frac{1}{\sqrt{q} - 1}$.

---

4 If we call $R_{AG}$ the quantity $1 - \delta - \frac{1}{\sqrt{q} - 1}$ we have the following theorem.

**Theorem 77** (Th 3.4 [42]) The Tsfasman–Vladut–Zink-bound

$R_{AG}$ lies completely below the Gilbert-Varshamov-bound for $q = p^{2^\nu} < 49$.

For $q = p^{2^\nu} \geq 49$ the two bounds intersect, and $R_{AG}$ lies above $R_{GV}$ on the interval $(\delta_1, \delta_2)$, where $\delta_1$ and $\delta_2$ are zeros of the equation

$$H_q(\delta) - \delta = (\sqrt{q} - 1)^{-1}.5$$

---

4 However the construction of Tsfasman-Vladut-Zink was extremely hard, and that because it does not give explicitly the concerned modular curves. Manin and Vladut have shown that these codes are actually polynomial constructable with an $O(n^{30})$ construction time the proofs in [42]Ch4.3. This complexity has since been reduced in $O(n^3)$.

5 We know that the random codes achieve the Gilbert-Varshamov bound with high
As we see from the previous theorem we have that bound $R_{AG}$ is interesting only for $q \geq 49$. Recently by using the $s-zeta$ function $Z^{(s)}(t) = \exp\left(\sum_{i=1}^{\infty} \frac{N_{i}}{s^{i}} t^{i}\right)$, of a projective smooth curve $C$, Xing in [49] has shown that

$$Z^{(s)}(t) = Z(t)(1 - t)^{s},$$

and then by using this result and by the construction of Goppa codes over a family $C$ such that $A_{q}(C) > 0$ by using special divisor $G$ which gives an improvement on the estimation of the parameters of the Goppa codes, he proved the following theorem.

**Theorem 78** (Th.4.2 [49] The Goppa Codes achieve asymptotically the Gilbert-Varshamov bound for any $\delta \in [0, 1 - 1/q]$.)

probability, however the result of the theorem above shows that there is a class of codes explicitly constructable which achieves the bound. This phenomena is quite rare in combinatorics. This is one of the primary reasons for the importance of the AG codes. Furthermore the AG codes has an efficient algorithm of decoding, whereas it is conjectured by the scientific community that the decoding of the random codes is NP-hard.

79
Chapter 7

List Decoding Algorithm for Goppa Codes

It is a natural ask to try to generalize the list decoding algorithm to the Goppa codes, since we have seen that they are a generalization of the RS codes. The problem was first treated by Shokrollahi and Wasserman in [32], later when Guruswami and Sudan gave the new algorithm in [14] they gave it also for a special case of Goppa codes, called one-point divisor codes\(^1\). All the results of this chapter have been taken from [15] with small modifications, some simplifications and remarks.

The definition of the one-point divisor codes is the same as our Goppa codes defined before except, they are \(C_L(D,G)\) with \(G = \alpha P_0\), where \(P_0\) is a place of degree 1 and \(\alpha\) a positive integer, \(K\) designe the function field \(\mathbb{F}_q(C)\) As before \(D = P_1 + \ldots + P_n\), where \(P_i\) are rational points such that \(P_0 \notin \{P_1, \ldots, P_n\}\). Then from the Theorem 69, we have that \(\dim(\mathcal{L}(\alpha P_0)) \geq \alpha-g+1\) and \(d \geq d^* = n-\alpha\), and if \(\alpha \geq 2g-1\), then \(\dim(\mathcal{L}(\alpha P_0)) = \alpha - g + 1\).

The problem of the list decoding for the Goppa codes is the same as the problem of RS codes i.e., for a given received word \(r = (r_1, \ldots, r_n) \in \mathbb{F}_q^n\).

\(^1\)The results can be easily generalized to the other Goppa codes.
the problem of finding the ball $B_\tau(r)$ of all words of $C_L(D,\alpha P_0)$ at distance at most $\tau$ of $r$ is equivalent to the problem of finding $B_\tau(r)^*$ of all rational functions $f \in \mathcal{L}(\alpha P_0)$, such that $f(P_i) = r_i$ for at least $n - \tau$ values of $i$. Formally we have to solve the following problem.

**Problem (Function Reconstruction: )**

**Input:** Integers $n, \alpha, \tau$, distinct pairs $\{(P_i, r_i)\}_{i=1}^n$

**Output:** All functions $f \in \mathcal{L}(\alpha P_0)$ such that $f(P_i) = r_i$ for at least $n - \tau$ values of $i \in [n]$.

### 7.1 Description of the Algorithm

The idea is the same as for the extended list-decoding algorithm for $RS$ codes.

Given data points $\{(P_i, r_i)\}_{i=1}^n$, we should find a polynomial $Q(y)$ in $K[y]$, such that

$$Q(y) = a_0 + \ldots + a_l y^l.$$ 

For $i \in [n]$ and $r_i \in \mathbb{F}_q$, $Q(r_i)$ yields an element of the function field $K = \mathbb{F}_q(C)$. Then since $Q(r_i)$ is a rational function in $K$ it can be evaluated at the place $P_i$. The idea is to require from our polynomial

$$Q(r_i)(P_i) \overset{\text{def}}{=} Q(P_i, r_i)$$

more, namely that the point $(P_i, r_i)$ behave likes a “zero of multiplicity $s$ of $Q$”. Having found the interpolation polynomial $Q$ by solving a homogeneous system of equations where the unknowns are the coefficients of $Q$. Then one finds the roots which are functions $f \in \mathcal{L}(\alpha P_0)$, which have distance at most $n - \tau$ from the received vector $r$. Formally we have to:

1. Pick parameters $s, \tau$ and $l, m$ suitably.
2. Find a non-zero polynomial $Q$ in $K[y]$, such that

$(i)$ the coefficients of the $y^i$ term in $Q \in K[y]$ belong in $\mathcal{L}((m - \alpha j)P_0)$

$(ii)$ for every $i \in \{1, 2, \ldots, n\}$ and $h$ in $K$, if $f(P_i) = y_i$, then

$$v_{P_i}(Q(f)) \geq s$$

3. Find all roots $f \in \mathcal{L}(\alpha P_0)$ of $Q \in K[y]$. For each of them check if $f(P_i) = r_i$ for at least $n - \tau$ values of $i$, and if so output $h$.

**A small remark on the parameter $l$:** Always from Riemann-Roch theorem we have

$$\dim \mathcal{L}((m - \alpha j)P_0) \geq m - \alpha j - g + 1 \text{ for all } 0 \leq j \leq l.$$ 

If we set $l \overset{def}{=} \left\lfloor \frac{m - g}{\alpha} \right\rfloor$, then we assure that $m - \alpha j - g + 1 \geq 1$.

For the condition $(i)$ on the polynomial, we assume that we have explicitly a basis $\phi_1, \ldots, \phi_{m-g+1}$ of $\mathcal{L}(mP_0)$ which satisfy

$$v_{P_0}(\phi_j) \geq -(j + g - 1),$$

(i.e., $\phi_j$ has at most $j + g - 1$ poles at $P_0$), and $v_{P_0}(\phi_j) > v_{P_0}(\phi_{j+1})$ for $1 \leq j \leq m - g + 1$ (i.e., the order at $P_0$ of $\phi_j$ increases by $j$).

The fact that $v_{P_0}(\phi_j) \geq -(j + g - 1) \implies \phi_j \in \mathcal{L}((j + g - 1)P_0)$.

The fact that $v_{P_0}(\phi_j) > v_{P_0}(\phi_{j+1})$ plus the fact that $\phi_j$ has at most $(j + g - 1)$ poles at $P_0$ implies that $\phi_j \in \mathcal{L}((j + g - 1)P_0) \subset \mathcal{L}((j + g)P_0)$ and $\phi_{j+1} \in \mathcal{L}((j + g)P_0) \setminus \mathcal{L}((j + g - 1)P_0)$. This implies the following:

for $1 \leq j_1 \leq m - g + 1 - \alpha j_2$, we have

$$\mathcal{L}((j_1 + g)P_0) \subset \mathcal{L}((m - \alpha j_2)P_0),$$

and $\phi_1, \ldots, \phi_{m-g+1-\alpha j_2}$ form a basis for $\mathcal{L}((m - \alpha j_2)P_0)$.
And then to satisfy the condition (i) we have to find the coefficients \( q_{j_1,j_2} \in \mathbb{F}_q \) such that the polynomial \( Q(y) \in K[y] \) can be written of the form

\[
Q(y) = \sum_{j_2=0}^{l} \sum_{j_1=1}^{m-g+1-\alpha_j} q_{j_1,j_2} \phi_{j_1} y^{j_2}.
\] (7.1)

By setting up \( Q \) as above the condition (i) is imposed.

To get the condition (ii) as for the RS codes we need to shift our basis. Since the points are over \( \mathcal{C} \) a different method is used. The following lemma is important for the condition (ii).

**Lemma 79** For every \( f, g \in K \) and a place \( P \) of degree one in \( \mathcal{C} \), such that \( v_P(f) = v_P(g) \), there exist \( \alpha_0, \beta_0 \in \mathbb{F}_q \setminus \{0\} \), such that \( v_P(\alpha_0 f + \beta_0 g) > v_P(f) \).

**Proof.** Let \( v_P(f) = v_P(g) = a \) and \( f^{-1} \) be the multiplicative inverse of \( f \) in \( K \). Then \( v_P(ff^{-1}) = 0 \) and \( v_P(f^{-1}) = -v_P(f) = -a \). Therefore, \( v_P(gf^{-1}) = v_P(g) + v_P(f^{-1}) = 0 \). Let \( f f^{-1}(P) = \alpha \) and \( gf^{-1}(P) = \beta \). We have \( \alpha, \beta \neq 0 \) then \( f \) has no pole at or zero at \( P \), and since the place \( P \) is rational over \( \mathcal{C} \) we have \( \alpha, \beta \in \mathbb{F}_q \). And then \( \beta ff^{-1} - \alpha gf^{-1} > 0 \) and so \( v_P(\beta f - \alpha g) > a \). Then it suffice to take \( \alpha_0 = \beta \) and \( \beta_0 = -\alpha \). \( \blacksquare \)

By using Lemma 79 we prove the following lemma.

**Lemma 80** Given non-zero functions \( \phi_1, \ldots, \phi_r \) of distinct pole orders at \( P_0 \) satisfying \( \phi_j \in \mathcal{L}((j + g - 1)P_0) \), and a place \( P_i \neq P_0 \), there exist non-zero functions \( \psi_1, \ldots, \psi_r \in K \) that satisfy:

(i) \( v_{P_i}(\psi_j) \geq j - 1 \) (i.e., \( \psi_j \) has at least a zero at \( P_i \) of order \( (j - 1) \)).

(ii) There exist \( \alpha_{P_i,j_1,j_3} \in \mathbb{F}_q \) for \( 1 \leq j_1, j_3 \leq r \) such that each function \( \phi_{j_1} \) can be expressed as a linear combination of the \( \psi_{j_3}'s \) of the from :

\[
\phi_{j_1} = \sum_{j_3=1}^{r} \alpha_{P_i,j_1,j_3} \psi_{j_3}.
\]
Proof. A stronger statement will be proved by induction on \( r \). If \( \phi_1, \ldots, \phi_r \) are linearly independent functions (over \( \mathbb{F}_q \)) that satisfy \( \nu_{P_i}(\phi_j) \geq t \) for each \( j = 1, 2, \ldots, r \), then there exist functions \( \psi_1, \ldots, \psi_r \) with \( \nu_{P_i}(\phi_j) \geq t + j - 1 \), that generate the \( \phi'_j \)'s over \( \mathbb{F}_q \). This will imply the result of the lemma, since the fact that the \( \phi'_j \)'s have distinct poles order at \( P_0 \) implies that \( \phi_1, \ldots, \phi_r \) are linearly independent. This follows from the fact that \( \nu_{P_0}(af + bg) = \min\{\nu_{P_0}(f), \nu_{P_0}(g)\} \), if \( \nu_{P_0}(f) \neq \nu_{P_0}(g) \).

The statement claimed is true for \( r = 1 \) with the choice \( \psi_1 = \phi_1 \). For \( r > 1 \) assume that \( \phi_1 \) is a function of the least zero order at \( P_i \). By assumption, \( \phi_1 \) has at least \( t \) zeroes at \( P_i \); i.e., \( \nu_{P_0}(\phi_1) \geq t \). We let \( \psi_1 = \phi_1 \). For \( 2 \leq j \leq r \), set \( \phi'_j = \phi_j \) if \( \nu_{P_i}(\phi_j) > \nu_{P_i}(\phi_1) \). Otherwise, if \( \nu_{P_i}(\phi_j) = \nu_{P_i}(\phi_1) \), using Lemma 79 to the pair \( (\phi_1, \phi_j) \), we get \( \alpha_j, \beta_j \in \mathbb{F}_q \setminus \{0\} \) such that the function \( \phi'_j = \alpha_j \phi_1 + \beta_j \phi_j \) satisfies \( \nu_{P_i}(\phi'_j) > \nu_{P_i}(\phi_1) \geq t \). Since \( \phi_j = \beta_j^{-1} \phi'_j - \alpha_j \beta_j^{-1} \phi_1 \) in this case, we conclude that in any case, for \( 2 \leq j \leq r \), \( \psi_1 = \phi_1 \) and \( \phi'_j \) generate \( \phi_j \). Then \( \phi'_2, \ldots, \phi'_r \) are linearly independents since \( \phi_1, \ldots, \phi_r \) are, and \( \nu_{P_i}(\phi'_j) \geq t + 1 \) for \( 2 \leq j \leq r \). Therefore the induction hypothesis applied to the functions \( \phi'_2, \ldots, \phi'_r \) now yields \( \psi_2, \ldots, \psi_r \) as required. \( \blacksquare \)

From the previous results we will express the condition \( (ii) \) on the polynomial \( Q(x, y) \). Using the lemma 80 and the equation (7.1) we deduce that \( Q \) has the form

\[
Q(y) = \sum_{j_2=0}^{1} \sum_{j_3=1}^{m-g+1} \sum_{j_1=1}^{m-g+1-j_2} q_{j_1,j_2} \alpha_{P_1,j_1,j_3} \psi_{j_3,P_1} y^{j_2}.
\]

The shifting to \( r_i \) is achieved by defining

\[
Q^{(i)}(y) = Q(y + r_i)
\]

The conditions on \( Q \) and \( P_i \) become
\( v_{P_i}(Q^{(i)}(f)) \geq s \) for all \( f \in K \) and such that \( f(P_i) = 0 \), i.e., \( \forall f \) s.t \( v_{P_i}(f) \geq 1 \).

Now

\[
Q^{(i)}(y) = \sum_{j_2=j_4}^{l} \sum_{j_1}^{m-g+1-\alpha j_2} \binom{j_2}{j_1} s^{j_2-j_4} q_{j_1,j_2,j_4} \alpha P_i \psi_{j_3,j_4} y^{j_4}. \tag{7.2}
\]

The terms in \( Q^{(i)} \) that are divisible by \( y^r \) contribute \( r \) towards the multiplicity of \((P_i, r_i)\) as a zero of \( Q \). Since \( v_{P_i}(\psi_{j_3,j_4}) \geq (j_3 - 1) \) by lemma 80, we can achieve the required condition on \( Q^{(i)} \) on \( Q^{(i)} \), or equivalently the required condition on \( Q \), by insisting that \( q_{j_3,j_4}^{(i)} = 0 \) for all \( j_3 \geq 1 \), \( j_4 \geq 0 \) such that \( j_4 + j_3 - 1 < s \), i.e. \( j_3 + j_4 \leq s \) (there are \( s+1 \) such constraints for each \( i \in \{1, 2 \ldots , n\} \)).

**Discussion on the parameters:** The above discussion shows that it is possible to solve the step 1 of the algorithm by finding a solution to a homogeneous system, with unknowns being the coefficients \( q_{j_1,j_2} \). The system has a non-zero solution if the number of unknowns which is at most \( \sum_{j_2=0}^{l} m - g + 1 - \alpha j_2 \) is greater than the number of constraints \( n \binom{s+1}{2} \). For that it suffices to set that

\[
\frac{(m-g)(m-g+2)}{2\alpha} > n \binom{s+1}{2}, \tag{7.3}
\]

and that comes from the fact that \( l = \lfloor \frac{m-g}{\alpha} \rfloor \) and then \( \sum_{j_2=0}^{l} m - g + 1 - \alpha j_2 \geq \frac{(m-g)(m-g+2)}{2\alpha} \). By replacing the values of \( m = s(n-\tau) - 1 \) in the inequality
7.3 the constraint becomes

\[ \frac{s(n - \tau - g)^2 - 1}{2\alpha} > n \left(s + \frac{1}{2}\right), \]

which simplifies to

\[ s^2[(n - \tau)^2 - \alpha n] - [2g(n - \tau) + \alpha n]s + (g^2 - 1) > 0. \]

If \((n - \tau)^2 - \alpha n > 0\), it suffices to pick \(s\) to be integer greater than the larger root of the above quadratic, and therefore picking

\[ s \overset{\text{def}}{=} 1 + \left\lfloor \frac{2g(n - \tau + \alpha n + \sqrt{[2g(n - \tau) + \alpha n]^2 - 4(g^2 - 1)(n - \tau)^2 - \alpha n])}}{2[(n - \tau)^2 - \alpha n]} \right\rfloor \]

suffices.

For the choice of the parameter \(m\) such that \(m \overset{\text{def}}{=} s(n - \tau) - 1\), in the last lemma of this chapter we will see that this choice is necessary to prove that the function in \(B_\tau(r)\) are in fact \(y\)-root of the polynomial \(Q\). Since we need \((n - \tau)^2 - \alpha n > 0 \iff \tau < n - \sqrt{\alpha n}\). This means that the algorithm list-decoding for AG codes corrects up to \(\tau\) errors, where

\[ \tau < n - \sqrt{\alpha n} = n - \sqrt{n(n - d^*)}. \]

### 7.1.1 Factorization Step in the List-Decoding Algorithm for AG Codes

To solve the step 2 of the algorithm list-decoding i.e., (find the \(y\)-roots) the idea is the same as for the factorization step in the list-decoding for the \(RS\) codes.

Let \(P\) be a place of \(K\) outside \(\text{supp} (\alpha P_0)\), that has degree \(r > \deg (\alpha P_0)\).
The following algorithm solves the step 2.

**Algorithm Root-Finding:**

1. Reduce the interpolation polynomial $Q$ modulo the place $P$ (compute $b_i = a_i(P)$ for all $0 \leq i \leq l$) and consider the polynomial

$$H(Y) = \sum_{i=0}^{l} b_i Y^i \in \mathbb{F}_{q^r}[y]$$

2. Compute the roots $\alpha_1, \ldots, \alpha_t$ of $H$ that lie in $\mathbb{F}_{q^r}$ using a root-finding algorithm for finite fields.

3. For each $\alpha_j, 1 \leq j \leq t$, find $\beta_j \in L(\alpha P_0)$ such that

$$\beta_j(P) = \alpha_j,$$ 

if any such $\beta_j$ exists.

For the RS codes we have seen the existence of an irreducible polynomial of degree $k$ over $\mathbb{F}_q$, which is used to reduce the polynomial, hence the correctness of the algorithm root-finding given in section 5.1. For AG codes it is not so clear that we can always find a place $P$ of some large degree. The following lemma assure this fact.

**Lemma 81** For any function field $K$, there exist a place for every large enough integer.

**Proof.** From the Hasse-Weil bound given in Theorem 72 and the remark 73, the number of places of degree $r$ on $C$ satisfies $|N_r - (q^r + 1)| \leq 2g\sqrt{q^r}$. Hence if $r \geq r_0$, where $r_0$ the smallest integer that satisfies $\frac{q^{r_0}-1}{2q^{r_0/2}} > g$, then $N_r \geq 1$.

Also it is not so obvious that in the root-finding algorithm after reducing the polynomial $Q$ modulo $P$, find the roots in $\mathbb{F}_q$ after doing the step 3 i.e., lifting the roots from $\mathbb{F}_{q^r}$ to $L(\alpha P_0)$ that “preserve” all the roots of
Q in 𝐿(α𝐏0). The following lemma prove the lifting is in fact an injective operation and hence preserve the roots.

**Lemma 82** Let 𝑓1, 𝑓2 ∈ 𝐿(𝐴) for some divisor 𝐴 ≥ 0, if 𝑓1(𝑃) = 𝑓2(𝑃), for some place 𝑃 with deg (𝑃) > deg (𝐴), then 𝑓1 = 𝑓2.

**Proof.** Suppose not, so that 𝑓1 − 𝑓2 ̸= 0. Then from remark 64 deg ((𝑓1 − 𝑓2)) = 0. But 𝑓1 − 𝑓2 ∈ 𝐿(𝐴) and 𝑓1(𝑃) − 𝑓2(𝑃) = 0 implies that 𝑣𝑃(𝑓1 − 𝑓2) ≥ 1, and then deg ((𝑓1 − 𝑓2)) ≥ deg (𝑃) − deg (𝐴) > 0. This is absurd. ■

7.1.2 Explicit List-Decoding Algorithm for AG Codes

With all the previous preparations we are now able to give the explicit list-decoding algorithm for AG codes.

**Input:** A received word (𝑟1, . . . , 𝑟𝑛), {𝐏0, . . . , 𝐿𝑛} n + 1 𝐹𝑞-rational points over a curve 𝐶.

**Parameters:** 𝑛, 𝑘, 𝜏

**Step 0:** Compute parameters 𝑠 such that

\[
\frac{(m - g)(m - g + 2)}{2\alpha} > n \binom{s + 1}{2}.
\]

With 𝑚 = 𝑠(𝑛 − 𝜏) − 1, in particular set

\[
s \equiv 1 + \left[ \frac{2g(n - \tau + \alpha n + \sqrt{(2g(n - \tau) + \alpha n)^2 - 4(g^2 - 1)((n - \tau)^2 - \alpha n)}}{2((n - \tau)^2 - \alpha n)} \right].
\]

**Step 1:** (Interpolation step) Find 𝑄(𝑦) ∈ 𝐿(𝑚𝐏0)[𝑦] of the form

\[
𝑄(𝑦) = \sum_{j_2=0}^{1} \sum_{j_1=1}^{m-g+1-\alpha j_2} q_{j_1,j_2} \phi_{j_1} y^{j_2}.
\]
for $l = \left\lceil \frac{m-g}{\alpha} \right\rceil$; i.e., find values of the coefficients $q_{j_1,j_2} \in \mathbb{F}_q$ such that the following holds:

1. At least one $q_{j_1,j_2}$ is non-zero (so that $Q$ is a non-zero polynomial in $K[y]$.)

2. For every $i \in [n]$ and for all $j_3 \geq 1, j_4 \geq 0$ such that $j_3 + j_4 \leq s$,

$$Q^{(i)}(y) = \sum_{j_2=j_4}^{l} \sum_{j_1}^{m-g+1-\alpha j_2} \binom{j_2}{j_4} r_i^{j_2-j_4} q_{j_1,jb} \alpha P_{i,j_1,j_3} = 0.$$ 

**Step 2:** (Root-finding step) Using the Root-Finding algorithm from the section, together with the place $P$ do the following:

(a) Find all roots $f \in L(\alpha P_0) \subseteq L(mP_0)$ of the polynomial $Q \in K[y]$.

(b) For each root, check if

$$f(P_i) = r_i$$

for at least $m$ values of $i$, and if so include $f$ in the output list.

**Step 3:** Output the list of all functions $f$ found in the Step 2.

We next prove that any $Q$ found in the interpolation step will have all the required functions $f$, namely those that satisfy $f(P_i) = r_i$ for at least $n - \tau$ values of $i$, as roots.

**Lemma 83** For $i \in \{1, \ldots, n\}$, if $f \in K$ satisfies $f(P_i) = r_i$, then

$$v_{P_i}(Q(f)) \geq s.$$
Proof. Using equation (7.1), we have the following for every place P:

\[ Q(f)(P) = \sum_{j_4=0}^{l} \sum_{j_3=1}^{m-g+1} q_{j_3,j_4}^{(i)}(\psi_{j_3,P_1}(P)(f(P) - r_i)^{j_4}. \quad (7.4) \]

And then for \( f(P_i) = r_i \), we get

\[ Q(f)(P) = \sum_{j_4=0}^{l} \sum_{j_3=1}^{m-g+1} q_{j_3,j_4}^{(i)}(\psi_{j_3,P_1}(P)(f(P) - f(P_i))^{j_4}. \]

By the constraints \( q_{j_3,j_4}^{(i)} = 0 \) for \( j_3 + j_4 \leq s \), and \( v_{P_1}(\psi_{j_3,P_1}) \geq j_3 - 1 \), and if \( f^{(i)} \) is defined by it value on places \( C \) as \( f^{(i)}(P) = f(P) - f(P_i) \), then \( v_{P_1}((f^{(i)})^{j_4}) \geq j_4 \). Then from equation 7.4 follows that \( v_{P_1}(Q(f)) \geq s \). As for Theorem 33 the following lemma shows the correctness of the list-decoding algorithm.

Lemma 84 If \( f \in \mathcal{L}(\alpha P_0) \) is such that \( f(P_i) = r_i \) for at least \( n - \tau \) values of \( i \in [n] \) and \( s(n - \tau) > m \), then \( Q(f) \equiv 0 \) i.e., \( h \) is a root of \( Q \in K[y] \).

Proof. By the choice of \( Q \) in the interpolation step, we have \( Q(f) \in \mathcal{L}(mP_0) \) for all \( f \in \mathcal{L}(\alpha P_0) \). Hence \( v_{P_1}Q(f) \geq 0 \) for each \( i \in [n] \). If \( f(P_i) = r_i \) for at least \( (n - \tau) \) values of \( i \), using Lemma 83, we get \( \sum_{i \in [n]} v_{P_1}(Q(f)) > m \), and hence the zero order of \( Q(f) \) is greater than \( m \). Since \( Q(f) \in \mathcal{L}(mP_0) \), the pole order of \( Q(f) \) is at most \( m \). Since there are more zeroes than poles for \( Q(f) \), and then by applying remark 64, we conclude that we must have \( Q(f) \equiv 0 \). Then \( f \) is a root of \( Q \).
Chapter 8

Conclusion

8.1 Contribution of this Work

He who enjoys doing what he has done is happy.

J.W. Goethe

The contribution of this work is concentrated in the following main aspects:

I Calculating the parameters of the non-primitive $BCH$ codes.

II A fine examination of the list-decoding algorithm for both the $RS$ codes and the AG codes, especially questions related to the parameters, the values of the errors tolerated as well as their bounds.

III Providing a new algorithm Root-Finding for the $RS$ codes.

8.1.1 The Parameters of the non-Primitive $BCH$ Codes

It is well known that it is difficult to find the dimension of the $BCH$ codes as we do not have the generator polynomial explicitly. Generally we have bounds. Our results are achieved by investigating the form of the cyclotomic classes (a simple tool, efficient but delicate to manipulate).
We gave an upper bound on the minimum distance of these codes, we compare it to the Griesmer bound. Note that in general the determination of the minimum distance is an \textit{NP}-Hard problem.

8.1.2 Presentation of the List-Decoding

The list-decoding algorithm was studied detailed and presented at length for:

- The \textit{RS} codes.
- The \textit{AG} codes.

8.1.3 The Solution for the Factorization Step in the list-Decoding Algorithm

We gave a property of the \textit{y}-roots which leads us to a new algorithm root-finding. The algorithm can be used for the both list-decoding algorithms.

8.2 Problems for Further Research

The results given in this work can be improved in the following sense.

- Find mathematically exactly or at least refine the bound on the minimum distance for our \textit{BCH} codes.
- The algorithm root-finding presented as solution for the factorization can be improved by finding other properties of the \textit{y}-roots.
Bibliography


93


