Universität Zürich[UZH]

b-Symbol Weight Distribution of Irreducible Cyclic Codes

Master Thesis

Blerim Alimehaj
18-719-427

University of Zurich
Institute of Mathematics

Supervisor: Prof. Dr. Joachim Rosenthal

Date: December 18, 2024

Abstract

As our technology advances, the need for b-symbol read channels that can handle messages with high-density data becomes crucial. The problem with conventional read channels is that they are more likely to overlap multiple information units, also called symbols, while reading messages with high-density data. The idea behind b-symbol read channels is that these channels consider all b consecutive symbols from the sent message as one symbol. This protects the message from being read with overlapping symbols. Considering a code word of some code $\mathcal{C}$ as the sent message, the message read by some b-symbol read channel is called a b-symbol code word.

In this thesis, we investigate b-symbol codes over semiprimitive irreducible cyclic codes and their Hamming weight distribution. The b-symbol Hamming weight distribution of semiprimitive irreducible cyclic codes is determined up to an invariant that we call $\mu(b)$ resp. $\mu_{(i)}(b)$. These invariants depend on b and on the choice of the primitive element that we use to describe the irreducible cyclic code $\mathcal{C}$. This thesis aims to find lower and upper bounds of the average values of these invariants. To do so, we use algebraic function field theory and number theory.

In this thesis, we obtain several lower and upper bounds, test their performance for smaller fields, and compare them. Furthermore, we are able to improve those bounds due to a number theoretic approach. Using these bounds, we are able to deduce a very good estimate for the average values of the invariants $\mu(b)$ resp. $\mu_{(i)}(b)$. These results provide a better understanding of the b-symbol Hamming weight distribution of semiprimitive irreducible cyclic codes.

Acknowledgements

First of all, I would like to thank Prof. Dr. Joachim Rosenthal, who, due to his lectures, introduced me to the exciting world of coding theory and cryptography. I would like to thank Prof. Dr. Ferruh Oezbudak for introducing me to b-symbol codes and for the useful discussions we had during my work. Without him, this thesis would not exist in this form. I would also like to thank Beatrice Toesca for her suggestions and feedback regarding my thesis. In addition, I want to thank all my university friends who helped me during my studies, especially Irena and Argetim, with whom I experienced many challenges and many pleasant moments. Lastly, I want to thank my family and my closest friends who supported me throughout my studies.

Contents

1 Introduction **1**

2 Preliminaries **3**
2.1 Field Theory . . . 3
2.1.1 Finite Fields . . . 3
2.1.2 Field Extensions . . . 5
2.1.3 Trace Map . . . 5
2.2 Algebraic Function Fields . . . 6
2.2.1 Basic Definitions and Properties of Algebraic Function Fields . . . 6
2.2.2 Algebraic Extensions of Function Fields . . . 9
2.2.3 Genus of a Function Field . . . 10
2.2.4 Zeta Function of a Function Field . . . 12
2.2.5 Hasse-Weil Bound . . . 13
2.3 Coding Theory . . . 14
2.3.1 Block Codes . . . 14
2.3.2 Linear Codes . . . 15
2.3.3 Cyclic Codes . . . 17

3 b-Symbol Codes **21**
3.1 Symbol-Pair Read Channel . . . 21
3.2 Cyclic Codes over Symbol-Pair Read Channels . . . 24
3.2.1 Pair Distance of Cyclic Codes . . . 25
3.2.2 Decoding Symbol-Pair Codes over Cyclic Codes . . . 26
3.3 b-Symbol Read Channel . . . 31

4 b-Symbol Weights of Cyclic Codes **35**
4.1 Trace Representation of Cyclic Codes . . . 35
4.2 Trace Representation of $\pi_b(\mathcal{C})$ . . . 38
4.3 b-Symbol Weights of Cyclic Codes . . . 42
4.3.1 Generalization of Weak Stability Theorem (Theorem 4.3.5) . . . 46
4.4 b-Symbol Weight Distribution of Semiprimitive Irreducible Cyclic Codes . . . 50
4.4.1 b-Symbol Weight Distribution of some Irreducible Cyclic Codes with $u = 2$ . . . 50
4.4.2 b-Symbol Weight Distribution of Semiprimitive Irreducible Cyclic Codes with arbitrary u . . . 59

5 Properties of the Sum of μ-Invariants of Symbol-Pair Codes. 65
5.1 Bounds of the Sum of μ-Invariants of Symbol-Pair Codes 67
5.1.1 Cardinalities of $\mathcal{A}$, $A^{(1)}$ and $A^{(-1)}$ 69
5.1.2 Bounds on $|\mathcal{M}|$ 72
5.1.3 Bounds on $|M^{(1)}|$ and $|M^{(-1)}|$ 76
5.2 Performance of LB and UB for Small q. 80
5.3 Divisors of the Sum of μ-Invariants 85

6 Conclusion 91

A Calculations of μ-Invariants and Bounds 93
A.1 Calculate the Sum of μ-Invariants 93
A.2 Calculate Lower and Upper Bounds of the Sum of μ-Invariants 95

Bibliography 110

Chapter 1

Introduction

An essential part of information theory is the recovery of messages that are going through channels. The information in those messages is contained and quantified in symbols over some alphabet. We can describe a piece of information as a vector $\mathbf{y} = (y_0, y_1, ..., y_{n-1}) \in \Sigma^n$, where $\mathbf{y}$ is a message of length n and $y_0, y_1, ..., y_{n-1}$ are symbols over some alphabet Σ. If $\mathbf{y}$ is sent and passes through a channel, then physical factors can corrupt the message. Therefore, it is crucial that the receiver has the possibility of recovering the original message. This is possible as long as the corrupted message does not exceed certain bounds. For instance, a block code with minimum Hamming distance d can correct up to $\lfloor (d-1)/2 \rfloor$ errors.
In some cases, the physical representation of the symbols differs from the way the channel itself reads those symbols. An example of where these problems occur is in the application of modern data storage technologies with high-density data where the channel may read multiple data symbols as one. In general, we have the problem that due to our technological progress and physical limitations, the used channel may not be able to read every single symbol. In 2010 and 2011, Y. Cassuto and M. Blaum thought of a way to overcome this issue by introducing symbol-pair read channels where the channel reads each pair of adjacent symbols as one information unit. This means that if $\mathbf{y} = (y_0, ..., y_{n-1}) \in \Sigma^n$ is the sent message, then the symbol-pair read channel reads the message $\pi_2(\mathbf{y}) = ((y_0, y_1), (y_1, y_2), ..., (y_{n-1}, y_0))$. One can extend the idea by introducing channels that read $b \geq 2$ adjacent symbols of a message as one information unit. These channels are called b-symbol read channels, and the messages read by such channels are called b-symbol read vectors. Considering code words as the initial message, one can define the corresponding b-symbol code words by identifying b adjacent symbols of a code word as one symbol. In other words, if $\mathbf{c} = (c_0, c_1, ..., c_{n-1})$ is a code word of some code $\mathcal{C}$, then the corresponding b-symbol code word is defined as

$$\pi_b(c) = \big((c_0, c_1, ..., c_{b-1}), (c_1, ..., c_b), ..., (c_{n-1}, c_0, ..., c_{b+n-2})\big),$$

where the indices are modulo n. The set of b-symbol code words over some code $\mathcal{C}$ is called the b-symbol code and denoted as $\pi_b(\mathcal{C})$.
In this thesis, we consider b-symbol codes over cyclic codes. We will determine the b-symbol weight distribution of semiprimitive irreducible cyclic codes up to an invariant, which we call $\mu(b)$ resp. $\mu_{(i)}(b)$. These invariants depend on the choice of the primitive element, which we use to describe the code $\mathcal{C}$. Since it is still an open problem to effectively calculate those invariants, and since the values of those invariants depend on the choice of the primitive

element, we will study the average of those values by considering the sum of each invariant over all primitive elements.
In Theorem 5.1.18, we will establish multiple lower and upper bounds for the sum of μ-invariants for semiprimitive irreducible cyclic codes under a simplified setting. The simplified setting consists of the following points:

- We consider symbol-pair codes, i.e., we set $b = 2$.
- $\mathcal{C}$ is a code of length n.
- $q = p^l$ where $p \neq 2$ is a prime and l is some positive integer.
- We consider fields of the form $\mathbb{F}_{q^4}$.
- $n \mid q^4 - 1$ such that $N = \frac{q^4-1}{n}$ satisfies $\gcd(\frac{q^4-1}{q-1}, N) = 2$.

We will also see that some bounds work for all fields $\mathbb{F}_{q^r}$ as long as $\gcd(\frac{q^r-1}{q-1}, N) = 2$ and r is even. We will test the bounds for primes $q \leq 43$ and observe that there is still room for improvement. Therefore, we will try to find useful properties of the sum of μ-invariants that can be used to improve these bounds. One idea is to find divisors. In Theorem 5.3.5, we will see that $2\varphi(q-1)$ divides the sum of μ-invariants, where $\varphi(\cdot)$ denotes the Euler phi function. This is a beneficial property that can be used to improve the bounds and in future proofs where divisors play an important role. The more we know about the invariants $\mu(b)$ resp. $\mu_{(i)}(b)$, the more we know about the b-symbol weight distribution of semiprimitive irreducible cyclic codes.
The thesis is structured as follows. In Chapter 2, we introduce the required background on finite fields, function field theory, and coding theory. In Chapter 3, we introduce the notations and basic properties of b-symbol codes. In Chapter 4, we determine the complete b-weight distribution of semiprimitive irreducible cyclic codes up to an invariant that we denote as $\mu(b)$ resp. $\mu_{(i)}(b)$. In Chapter 5, we investigate the sum of the invariants $\mu(b)$ resp. $\mu_{(i)}(b)$ under a simplified setting to establish useful properties. We determine lower and upper bounds of the sum of μ-invariants and eventually test those bounds, and we will see that some bounds have advantages over others. In addition, we will find divisors of the sum of μ-invariants, which can be used to improve the lower and upper bounds and which are also beneficial in proofs where divisors play an important role.

Chapter 2

Preliminaries

In this Chapter, we build the foundation of our future work. First, we will give the main definitions and properties of finite fields. After that, we introduce the essential background of algebraic function field theory. Lastly, we will provide a short introduction to coding theory, which allows us to introduce b-symbol codes and our main issue, which is finding the b-symbol Hamming weight distribution of some irreducible cyclic codes.

2.1 Field Theory

This section will focus on finite fields and provide the main definitions and properties we need for our work. A much more profound view on this topic can be found in [9], [11], and [13].

2.1.1 Finite Fields

Definition 2.1.1 (Field)
A field is a set $\mathbb{F}$ equipped with two binary operations $+$ and $\times$ satisfying the following properties.

- *$\mathbb{F}$ is an abelian group under $+$ with identity element 0.*
- *$\mathbb{F} \setminus \{0\}$ is an abelian group under $\times$ with identity element 1.*
- *$\times$ satisfy distributivity over $+$, i.e. $a \times (b + c) = (a \times b) + (a \times c)$ for all a, b, c in $\mathbb{F}$.*

Throughout this work, we will always refer to $\mathbb{F}$ as a field and we will use the terminology $\mathbb{F}^*$ for the nonzero elements of the field $\mathbb{F}$.
We call the number of elements in a field $\mathbb{F}$ the order of $\mathbb{F}$. If the order of $\mathbb{F}$ is finite, we call $\mathbb{F}$ a finite field. E.g., $\mathbb{Z}/(p)$, the ring of residue classes of the integer modulo the principal ideal generated by a prime p, is a finite field of order p [9].

Definition 2.1.2 (Characteristic)
Let $\mathbb{F}$ be a field. If a positive integer n exists, such that $na = 0$ for any a in $\mathbb{F}$, then the smallest integer n with this property is called the characteristic of $\mathbb{F}$.
If no such n exists, then we say that the characteristic of $\mathbb{F}$ is 0.

Theorem 2.1.3 (Order of a Finite Field [11])
Let $\mathbb{F}$ be a finite field. Then $\mathbb{F}$ has p^n elements, where p is a prime and the characteristic of $\mathbb{F}$, and n is the degree of $\mathbb{F}$ over its prime subfield, which is the subfield of $\mathbb{F}$ of prime order p.

One can easily construct any finite field of order p^n by considering the polynomial ring $\mathbb{F}[x]$ and any irreducible polynomial f in $\mathbb{F}[x]$. The residue class ring $\mathbb{F}[x]/(f)$ is a field if and only if f is irreducible over $\mathbb{F}[x]$. Moreover, if p is the characteristic of $\mathbb{F}$, and n is the degree of f, then $\mathbb{F}[x]/(f)$ is a field of order p^n. The next theorem tells us that all fields of order p^n constructed in this way are isomorphic.

Theorem 2.1.4 (Existence and Uniqueness of Finite Fields [11])
For every prime p and every positive integer n, there exists a finite field with p^n elements. Any finite field with $q = p^n$ elements is isomorphic to the splitting field of $x^q - x$ over $\mathbb{Z}/(p)$.

From now on, we will denote a finite field as $\mathbb{F}_q$, where $q = p^n$ for some prime p and positive integer n. Considering fields of the form $\mathbb{F}_q$, there is a valuable subfield criterion to determine the subfields of $\mathbb{F}_q$.

Theorem 2.1.5 (Subfield Criterion [11])
Let $\mathbb{F}_q$ be the finite field with $q = p^n$ elements. Then every subfield of $\mathbb{F}_q$ has order p^m, where m is a positive divisor of n. Conversely, if m is a positive divisor of n, then there is exactly one subfield of $\mathbb{F}_q$ with p^m elements.

For any finite field $\mathbb{F}_q$, we call $\mathbb{F}_q^*$, the multiplicative group of nonzero elements of $\mathbb{F}_q$. Further, the multiplicative group $\mathbb{F}_q^*$ is cyclic.

Theorem 2.1.6 ($\mathbb{F}_q^*$ is Cyclic [11])
For every finite field $\mathbb{F}_q$, the multiplicative group $\mathbb{F}_q^*$ of nonzero elements of $\mathbb{F}_q$ is cyclic.

Since $\mathbb{F}_q^*$ is cyclic, we know that there is a generator $\eta \in \mathbb{F}_q$ such that

$$\mathbb{F}_q^* = \langle \eta \rangle = \{\eta^0, \eta^1, ..., \eta^{p^n-2}\}.$$

Definition 2.1.7 (Primitive Element)
If $\eta \in \mathbb{F}_q$ is a generator of $\mathbb{F}_q^*$, then we call η a primitive element of $\mathbb{F}_q$.

One can ask the question, how many primitive elements are there? We will need the Euler phi function (also called the Euler's totient function) to answer this.

Definition 2.1.8 (Euler's phi Function)
For any integer n, the Euler phi function $\varphi(n)$ indicates the number of integers $1 \leq i \leq n$ such that $gcd(i, n) = 1$. Here, $gcd(i, n)$ denotes the greatest common divisor of i and n.

To determine the number of primitive elements, we use the fact that for any primitive element $\eta \in \mathbb{F}_q$, η^i is a primitive element if and only if $gcd(i, q-1) = 1$ [11].

Lemma 2.1.9 (Number of Primitive Elements [11])
For any field $\mathbb{F}_q$, the number of primitive elements is $\varphi(q-1)$.

The Euler phi function will be substantial in this thesis since we will investigate special sums over all primitive elements. Therefore, by multiplying the smallest or biggest term within the sum with the number of primitive elements, we will obtain some simple upper and lower bounds.

2.1.2 Field Extensions

Definition 2.1.10 (Field Extension)
Let L be a field and K be a subfield of L. Then, L/K is called a field extension.

Further, if L is a finite field. We can consider L as K-vector space, call its dimension the degree of L/K, and denote it by $[L : K]$. The most important field extensions for our purpose will be the case where $L = \mathbb{F}_{q^r}$ and $K = \mathbb{F}_q$ for some positive integer r. In this case, the degree of $\mathbb{F}_{q^r}/\mathbb{F}_q$ is $[\mathbb{F}_{q^r} : \mathbb{F}_q] = r$.
Let $Aut(L/K)$ denote the group of automorphisms of L that fix K. This means that any element $\sigma \in Aut(L/K)$ is a K-isomorphism from L onto L.

Definition 2.1.11 (Galois Extension)
A field extension L/K is said to be a Galois extension if the order of $Aut(L/K)$ is $[L : K]$. In this case, we denote the group of automorphisms of L over K by $Gal(L/K) := Aut(L/K)$.

A more detailed view of field extensions and Galois extensions can be found in textbooks on algebra as in [11]. In the case where $L = \mathbb{F}_{q^r}$ and $K = \mathbb{F}_q$ we have that $\mathbb{F}_{q^r}/\mathbb{F}_q$ is a Galois extension.

Theorem 2.1.12 ($\mathbb{F}_{q^r}/\mathbb{F}_q$ is a Galois Extension [11])
The field extension $\mathbb{F}_{q^r}/\mathbb{F}_q$ is a Galois extension. We denote its Galois group by $Gal(\mathbb{F}_{q^r}/\mathbb{F}_q)$.

2.1.3 Trace Map

The trace map will be crucial in this work, and we will use it especially when investigating trace codes. For now, we want to provide the definition and some important properties.

Definition 2.1.13 (Trace Map [9])
Let $\alpha \in F = \mathbb{F}_{q^m}$ and $K = \mathbb{F}_q$, then the trace of α over K is defined by

$$Tr_{F/K} : F \to K,\ Tr_{F/K}(\alpha) = \alpha + \alpha^q + \cdots + \alpha^{q^{m-1}}$$

We usually denote the trace map by $Tr(\cdot)$ if F and K are clear. An essential property of the trace map is its linearity.

Theorem 2.1.14 (Properties of the Trace Map [9])
Let $K = \mathbb{F}_q$ and $F = \mathbb{F}_{q^m}$. Then, the trace map satisfies the following properties:

(i) $Tr_{F/K}(\alpha + \beta) = Tr_{F/K}(\alpha) + Tr_{F/K}(\beta)$ *for all* $\alpha, \beta \in F$;

(ii) $Tr_{F/K}(c\alpha) = cTr_{F/K}(\alpha)$ *for all* $c \in K$, $\alpha \in F$;

(iii) $Tr_{F/K}$ *is a linear transformation from F onto K, where both F and K are viewed as vector spaces over K;*

(iv) $Tr_{F/K}(a) = ma$ *for all* $a \in K$;

(v) $Tr_{F/K}(\alpha^q) = Tr_{F/K}(\alpha)$ *for all* $\alpha \in F$.

Using the linearity of the trace map, we will later obtain some linear codes, which we will refer to as trace codes or trace representations of linear codes. Notice that the properties (i) and (v) provide us with a crucial fact about the zeros of the trace map. Namely, that

$$Tr_{F/K}(x) = 0 \iff y^q - y = x \quad \text{for some } y \in \mathbb{F}_{q^m}.$$

The zeros of the trace map will be essential for this work since we are interested in the Hamming weights of trace codes. Trace codes have useful properties, which we will see in the next sections. Before that, we also introduce some background on algebraic function fields.

2.2 Algebraic Function Fields

This section will introduce the background of algebraic function fields we need for our work. The Hasse-Weil bound will especially be helpful, as it will give us an upper bound for the number of rational places of some algebraic function field. A more detailed view on this topic can be found in [18].

2.2.1 Basic Definitions and Properties of Algebraic Function Fields

For this section, we will denote an arbitrary field by K. Recall that a rational function field is defined as

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

Definition 2.2.1 (Algebraic Function Field)
An algebraic function field F/K of one variable over K is a field extension $F \supseteq K$ such that F is a finite algebraic extension of $K(x)$ for some element $x \in F$ which is transcendental over K. I.e., there is no $f \in K[x]$ such that $f(x) = 0$.

Example 2.2.2 (Rational Function Field [18])
The simplest example of an algebraic function field is the rational function field. F/K is rational if $F = K(x)$ for some $x \in F$ which is transcendental over K. For all non-zero $z \in K(x)$, we have a unique representation by

$$z = a \cdot \prod_i p_i(x)^{n_i},$$

where $a \in K \setminus \{0\}$, $n_i \in \mathbb{Z}$ and the polynomials $p_i(x) \in K[x]$ are irreducible, monic and pairwise distinct.

Definition 2.2.3 (Constant Field)
Let $\tilde{K} := \{z \in F \ : \ z \text{ is algebraic over } K\}$. Then, since sums, products, and inverses of algebraic elements are also algebraic, $\tilde{K}$ is a subfield of F. We call $\tilde{K}$ the constant field of F or just the constant field.

For our convenience, since we work only over finite fields, we make some assumptions that hold for this entire work.

- K is a finite field.

- We denote F/K, or just F if K is clear, the algebraic function field F over K.
- $K = \tilde{K}$, i.e., every element in F that is algebraic over K is in K.

We can represent F/K as a simple algebraic field extension of a rational function field $K(x)$. For instance, if $[F : K(x)] = m$, then there exists an irreducible polynomial

$$h(T) = h_0 + h_1 T + \cdots + h_m T^m \in K(x)[T]$$

with $h(y) = 0$ such that $F = K(x, y) = K(x)[y]/ < h(y) >$.

Definition 2.2.4 (Valuation Ring)
A valuation ring of the function field F/K is a ring $\mathcal{O} \subseteq F$ with the following properties:

i) $K \subsetneq \mathcal{O} \subsetneq F$.

ii) *For every $z \in F$ we have either $z \in \mathcal{O}$ or $z^{-1} \in \mathcal{O}$.*

The following example illustrates well why the definition 2.2.4 is useful.

Example 2.2.5 ([18])
Assume F/K is the rational function field, $F = K(x)$. Let $p(x) \in K[x]$ be a monic irreducible polynomial. Consider the set

$$\mathcal{O}_{p(x)} := \left\{ \frac{f(x)}{g(x)} \ : \ f(x), g(x) \in K[x],\ p(x) \nmid g(x) \right\}.$$

One can easily verify that $\mathcal{O}_{p(x)}$ is a valuation ring of $K(x)/K$. Moreover, if $q(x) \neq p(x)$ is another irreducible monic polynomial in $K[x]$, then $\mathcal{O}_{q(x)} \neq \mathcal{O}_{p(x)}$.

We denote the group of units of $\mathcal{O}$ as

$$\mathcal{O}^\times = \{u \in \mathcal{O} \ : \ \exists v \in \mathcal{O} \ \text{ such that } uv = 1\}.$$

The next proposition shows that the valuation ring $\mathcal{O}$ is a local ring. In other words, $\mathcal{O}$ has a unique maximal ideal $P = \mathcal{O} \setminus \mathcal{O}^\times$.

Proposition 2.2.6 ($\mathcal{O}$ is a Local Ring [18])
Let $\mathcal{O}$ be a valuation ring of the function field F/K. Then the following hold:

i) *$\mathcal{O}$ is a local ring; i.e., $P = \mathcal{O} \setminus \mathcal{O}^\times$ is the unique maximal ideal of $\mathcal{O}$.*

ii) *Let $0 \neq x \in F$. Then $x \in P \iff x^{-1} \notin \mathcal{O}$.*

iii) *For the constant field K of F/K we have $K \subseteq \mathcal{O}$ and $K \cap P = \{0\}$.*

Definition 2.2.7 (Place)
A place P of the function field F/K is the maximal ideal of some valuation ring $\mathcal{O}$ of F/K.

Notice that Proposition 2.2.6 *ii)* implies that $\mathcal{O}$ is uniquely determined by P through $\mathcal{O} = \{z \in F \ : \ z^{-1} \notin P\}$. Therefore, we denote the valuation ring of the place P as $\mathcal{O}_P$. Since P is the maximal ideal of $\mathcal{O}_P$, the residue class ring $F_P = \mathcal{O}_P/P$ is a field. We call F_P the residue class field of the place P. Proposition 2.2.6 *iii)* implies that the residue class map $\mathcal{O}_P \to F_P = \mathcal{O}_P/P$ induces a canonical embedding of K into F_P. We conclude that K is a subfield of F_P. Therefore, we can discuss the extension degree of F_P over K, i.e., $[F_P : K]$.

Definition 2.2.8 (Degree of Place)
Let F_P be the residue class field of the place P. Then we call the extension degree of F_P over K the degree of P, and we write

$$\deg P := [F_P : K].$$

Furthermore, the degree of a place is always finite [18]. If $\deg P = 1$, we call P a rational place of F/K.

Example 2.2.9 ([17])
Let $K = \mathbb{F}_q$ and $F = \mathbb{F}_q(x)$, the rational function field over $\mathbb{F}_q$. There exist exactly $q+1$ places of degree one (rational places) of F and they are given as follows:

i) For $\alpha \in \mathbb{F}_q$, let

$$P_\alpha = \left\{ \frac{a(x)}{b(x)} \ : \ a(\alpha) = 0,\ b(\alpha) \neq 0 \right\}, \tag{2.1}$$

where $a(x), b(x) \in \mathbb{F}_q[x]$. This gives us q (affine) rational places of F.

ii) There is one rational place at infinity of F. It is given by

$$P_\infty = \left\{ \frac{a(x)}{b(x)} \ : \ \deg a(x) < \deg b(x) \right\}, \tag{2.2}$$

where $a(x), b(x) \in \mathbb{F}_q$.

These are all rational places of F. In general, places of degree $m \geq 1$ different from P_∞ are given as follows. Let $r(x) \in \mathbb{F}_q[x]$ be an irreducible polynomial of degree m. Then

$$P_{r(x)} = \left\{ \frac{a(x)}{b(x)} \ : \ r(x) \mid a(x),\ r(x) \nmid b(x) \right\}, \tag{2.3}$$

where $a(x), b(x) \in \mathbb{F}_q[x]$. Notice that for $m = 1$, the notations (2.1) and (2.3) are equivalent, i.e., $P_\alpha = P_{x-\alpha}$.

Definition 2.2.10 (Local Parameter of P)
Let P be the place corresponding to the valuation ring $\mathcal{O}$. Every element $t \in P$ such that $P = t\mathcal{O}$ is called a local parameter of P (other notations are prime element or uniformizing variable of P).

Definition 2.2.11 (Discrete Valuation v_P)
Let P be a place of F and $\mathcal{O}_P$ its corresponding valuation ring. We define the discrete valuation map v_P by choosing a local parameter t of P and mapping the elements as follows: Every element $0 \neq z \in F$ has a unique representation $z = t^n u$, where $n \in \mathbb{Z}$ and $u \in \mathcal{O}_P^\times$. We define v_P as

$$\begin{aligned} v_P : F &\to \mathbb{Z} \cup \{\infty\} \\ 0 \neq z = t^n u &\mapsto n, \\ 0 &\mapsto \infty. \end{aligned}$$

Notice that the definition of v_P only depends on the place P and not on the choice of the local parameter t. Indeed, if t' is another local parameter of P, then $P = t\mathcal{O}_P = t'\mathcal{O}_P$. Therefore, we can find $w \in \mathcal{O}_P^\times$ such that $t = t'w$. This implies that for any non-zero $z = t^n u \in F$, we have $t^n u = t'^n w^n u = t'^n (w^n u)$ with $w^n \in \mathcal{O}_P^\times$ and $v_P(t^n u) = v_P(t'^n (w^n u)) = n$.

2.2.2 Algebraic Extensions of Function Fields

We assume, as previously, that $K = \tilde{K}$ and that K is finite.

Definition 2.2.12 (Algebraic Extension of a Function Field)
An algebraic function field F'/K' is called an algebraic extension of F/K if $F' \supseteq F$ is an algebraic field extension and $K' \supseteq K$.

If further $F' = FK'$ is the composite field of F and K', then we call the algebraic extension F'/K' of F/K a constant field extension. For the next definition, we denote the set of places of the function field F/K as

$$\mathbb{P}_F := \{P \ : \ P \ \text{ is a place of } \ F/K\}$$

Definition 2.2.13 (Extension of Places)
Let F'/K' be an algebraic extension of F/K. A place $P' \in \mathbb{P}_{F'}$ is called an extension of $P \in \mathbb{P}_F$ if $P \subseteq P'$, and we write $P'|P$.

Proposition 2.2.14 ([18])
Let F'/K' be an algebraic extension of F/K. Suppose that $P \in \mathbb{P}_F$ and $P' \in \mathbb{P}_{F'}$, and let $\mathcal{O}_P \subseteq F$, $\mathcal{O}_{P'} \subseteq F'$ be the corresponding valuation ring, and v_P (resp. $v_{P'}$) the corresponding discrete valuation. Then the following are equivalent:

1. *$P'|P$.*
2. *$\mathcal{O}_P \subseteq \mathcal{O}_{P'}$.*
3. *There exists an integer $e \geq 1$ such that $v_{P'}(x) = e \cdot v_P(x)$ for all $x \in F$.*

Moreover, if $P'|P$ then $P = P' \cap F$ and $\mathcal{O}_P = \mathcal{O}_{P'} \cap F$. Therefore, P is also called the restriction of P' to F. A consequence of Proposition 2.2.14 is that we have a canonical embedding of the residue class field $F_P = \mathcal{O}_P/P$ into $F'_{P'} = \mathcal{O}_{P'}/P'$. This implies we can view F_P as a subfield of $F'_{P'}$.

Definition 2.2.15 (Ramification Index)
Let F'/K' be an algebraic extension of F/K, and let $P' \in \mathbb{P}_{F'}$ be an extension of $P \in \mathbb{P}_F$. The integer $e(P'|P) := e$ with

$$v_{P'}(x) = e \cdot v_P(x) \ \text{ for all } x \in F$$

is called the ramification index of P' over P. We say that $P'|P$ is ramified if $e(P'|P) > 1$, and $P'|P$ is unramified if $e(P'|P) = 1$.

Since F_P is a subfield of $F_{P'}$ we can define the relative degree of P' over P as

$$f(P'|P) := [F'_{P'} \ : \ F_P].$$

We obtain the fundamental equality theorem by combining the ramification index with the relative degree of $P'|P$.

Theorem 2.2.16 (Fundamental Equality [18])
Let F'/K' be a finite extension of F/K. Let $P \in \mathbb{P}_F$ and let $P_1, P_2, ..., P_m$ be all the place extensions of P, i.e. $P_i \cap F = P$ for all $1 \leq i \leq m$. Let $e_i := e(P_i|P)$ denote the ramification index and $f_i := f(P_i|P)$ the relative degree of $P_i|P$. Then

$$\sum_{i=1}^{m} e_i \cdot f_i = [F' \ : \ F].$$

2.2.3 Genus of a Function Field

The genus is one of the most important invariants of a function field. We will need the genus to state the Hasse-Weil bound for rational places (places of degree one). We will start by defining the divisor group of a function field F/K. As previously mentioned, F/K refers to a function field with a full constant field K.

Definition 2.2.17 (Divisor Group of F/K ($Div(F)$))
The divisor group of F/K is defined as the (additively written) free abelian group generated by the places of F/K. We will denote the group by $Div(F)$. The elements of $Div(F)$ are called divisors of F/K, i.e. any element $D \in Div(F)$ can be written as

$$D = \sum_{P \in \mathbb{P}_F} n_P \cdot P \quad \textit{with} \quad n_P \in \mathbb{Z}, \quad \textit{and almost all} \quad n_P = 0.$$

The support of D is defined as

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

We will now state some basic properties of divisors.

- For $D \in Div(F)$, if $D = P$ with $P \in \mathbb{P}_F$, then we call D a prime divisor.
- Let $D = \sum n_P P$ and $D' = \sum n'_P P$, then we can obtain the divisor $D + D'$ by adding the coefficients, i.e.,
$$D + D' = \sum_{P \in \mathbb{P}_F} (n_P + n'_P) P.$$
- We have $0 \in Div(F)$ as the zero element of the divisor group and define it as
$$0 := \sum_{P \in \mathbb{P}_F} r_P P, \quad \text{where all } r_P = 0.$$
- For $Q \in \mathbb{P}_F$ and $D = \sum n_P P \in Div(F)$, we define $v_Q(D) := n_Q$. Therefore, we also can write the support of D as
$$supp(D) = \{P \in \mathbb{P}_F \ : \ v_P(D) \neq 0\} \ \text{ and } \ D = \sum_{P \in supp(D)} v_P(D) \cdot P.$$
- We can define a partial ordering on $Div(F)$ by
$$D_1 \leq D_2 :\Longleftrightarrow v_P(D_1) \leq v_P(D_2) \ \text{ for all } P \in \mathbb{P}_F.$$
- We can define the degree of a divisor by
$$\deg D := \sum_{P \in \mathbb{P}_F} v_P(D) \cdot [F_P \ : \ K] = \sum_{P \in \mathbb{P}_F} v_P(D) \cdot \deg P.$$

It is known that in a function field F/K, every non-zero element in F has only finitely many zeros and poles [18]. This fact motivates the following definition.

Definition 2.2.18 (Zero-, Pole-, Principal Divisor)
Let $0 \neq x \in F$ and let Z (resp. N) be the set of zeros (resp. poles) of x in $\mathbb{P}_F$. We define

$$(x)_0 := \sum_{P \in Z} v_P(x)P, \quad \textit{the zero divisor of } x$$
$$(x)_\infty := \sum_{P \in N} (-v_P(x))P, \quad \textit{the pole divisor of } x$$
$$(x) := (x)_0 - (x)_\infty, \quad \textit{the principal divisor of } x.$$

Using principal divisors, we are ready to define Riemann-Roch spaces, which play a fundamental role in the theory of algebraic function fields.

Definition 2.2.19 (Riemann-Roch Space)
Let $A \in Div(F)$. We define the Riemann-Roch space associated to the divisor A by

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

One can view the elements of $\mathscr{L}(A)$ as follows: if

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

with $n_i > 0$ and $m_j > 0$. Then $\mathscr{L}(A)$ consists of all elements $x \in F$ such that

- x has zeros of order $\geq m_j$ at Q_j, for $j = 1, 2, ..., s$, and
- x may have poles only at places $P_1, P_2, ..., P_r$, with the pole order at P_i bounded by n_i for $i = 1, 2, ..., r$.

It is a well-known fact that the Riemann-Roch space is indeed a vector space over K. Furthermore, for any divisor $A \in Div(F)$, we have that $\mathscr{L}(A)$ is a finite-dimensional vector space over K [18]. We denote the dimension of $\mathscr{L}(A)$ over K as

$$\ell(A) := \dim(\mathscr{L}(A)).$$

Proposition 2.2.20 ([18])
There exists a constant $\gamma \in \mathbb{Z}$ such that for all divisors $A \in Div(F)$ we have the following bound:

$$\deg A - \ell(A) \leq \gamma.$$

This implies that there exists an integer g, which only depends on the function field F/K such that for all divisors $A \in Div(F)$

$$g \geq \deg A - \ell(A) + 1.$$

By setting g as the maximum of the right-hand side over all divisors A, we obtain the most important invariant of a function field [18].

Definition 2.2.21 (Genus)
The genus of the function field F/K is defined by

$$g := \max\{\deg A - \ell(A) + 1 \ : \ A \in Div(F)\}.$$

2.2.4 Zeta Function of a Function Field

Let F be an algebraic function field of genus g with constant field $\mathbb{F}_q$. Let $A \in Div(F)$. We write A as

$$A = \sum_{P \in \mathbb{P}_F} a_P P.$$

We say that A is positive if all $a_P \geq 0$ and write $A \geq 0$. It is a known fact that for every $n \geq 0$, only finitely many positive divisors of degree n exist [18]. Therefore, we can define the following positive integers

$$A_n := |\{A \in Div(F) \ : \ A \geq 0, \ \deg A = n\}|. \tag{2.4}$$

Using these integers, we can associate each function field $F/\mathbb{F}_q$ with the following power series

Definition 2.2.22 (Zeta Function of $F/\mathbb{F}_q$)
Let A_n be as in equation (2.4). Then we define the Zeta function of $F/\mathbb{F}$ by

$$Z(t) := Z_F(t) := \sum_{n=0}^{\infty} A_n t^n \in \mathbb{C}[[t]].$$

The Zeta function contains a lot of information about the corresponding function field. Furthermore, it is part of a function field's L-polynomial.

Definition 2.2.23 (L-Polynomial of $F/\mathbb{F}_q$)
The L-polynomial of $F/\mathbb{F}_q$ is defined by

$$L(t) := L_F(t) := (1-t)(1-qt)Z(t).$$

The L-polynomial has many useful properties. For instance, the L-polynomial factors in $\mathbb{C}[t]$.

Theorem 2.2.24 (Properties of $L(t)$ [18])
Let $L(t)$ be the L-polynomial of $F/\mathbb{F}_q$. Then $L(t)$ satisfies the following properties:

- $L(t) \in \mathbb{Z}[t]$ *and* $\deg L(t) = 2g$.
- $L(t)$ *factors in* $\mathbb{C}[t]$ *as*

$$L(t) = \prod_{i=1}^{2g} (1 - \alpha_i t). \tag{2.5}$$

Let $F_r = F\mathbb{F}_{q^r}$ be the constant field extension of $F/\mathbb{F}_q$ of degree r. Then its L-polynomial $L_r(t)$ factors similarly to (2.5) in $\mathbb{C}[t]$ in the form

$$L_r(t) = \prod_{i=1}^{2g} (1 - \alpha_i^r t). \tag{2.6}$$

Using equation (2.5), we can define $\alpha_1, \alpha_2, ..., \alpha_{2g}$ as the reciprocals of the zeros of $L(t)$. These reciprocals appear to be very useful in determining the number of rational places of a function field. We denote the number of rational places as follows:

$$N(F) := |\{P \in \mathbb{P}_F \ : \ \deg P = 1\}|.$$

Corollary 2.2.25 ([18])
For all $r \geq 1$,

$$N(F_r) = q^r + 1 - \sum_{i=1}^{2g} \alpha_i^r,$$

where $\alpha_1, ..., \alpha_{2g} \in \mathbb{C}$ are the reciprocals of the zeros of $L(t)$.

Furthermore, the Hasse-Weil theorem ([18], Theorem 5.2.1) states that the absolute values of all α_i in $L_F(t)$ is always $q^{1/2}$.

Theorem 2.2.26 (Hasse-Weil [18])
The reciprocals of the zeros of $L_F(t)$ satisfy

$$|\alpha_i| = q^{1/2} \quad \text{for} \quad i = 1, 2, ..., 2g. \tag{2.7}$$

Using Corollary 2.2.25 and Theorem 2.2.26, one can deduce very useful bounds on $N(F)$. One of them is the Hasse-Weil bound.

2.2.5 Hasse-Weil Bound

Let $K = \mathbb{F}_q$. We consider the function field $F/\mathbb{F}_q$ with genus $g(F) = g$. We use the following notations

- $N(F) = |\{P \in \mathbb{P}_F \ : \ \deg P = 1\}|$, i.e., $N(F)$ is the number of rational places of $F/\mathbb{F}_q$.
- $F_r = F \cdot \mathbb{F}_{q^r}$ is the constant field extension of degree r.
- $N_r = N(F_r)$.

From Corollary 2.2.25 and Theorem 2.2.26, one can deduce immediately the Hasse-Weil bound.

Theorem 2.2.27 (Hasse-Weil Bound [18])
The number of rational places $N = N(F)$ of $F/\mathbb{F}_q$ satisfies the inequality

$$|N - (q+1)| \leq 2gq^{1/2}.$$

Applying Theorem 2.2.27 to $F_r/\mathbb{F}_{q^r}$ we obtain the inequality

$$|N_r - (q^r + 1)| \leq 2gq^{r/2}.$$

We will usually use the inequality for $r = 1$. Furthermore, it gives us upper and lower bounds for rational places by

$$q + 1 - 2gq^{1/2} \leq N \leq q + 1 + 2gq^{1/2}.$$

Definition 2.2.28 (Maximal and Minimal Function Field)
A function field $F/\mathbb{F}_q$ of genus g is called maximal (resp. minimal) if

$$N = q + 1 + 2gq^{1/2} \ (\ \textit{resp.} \ \ N = q + 1 - 2gq^{1/2}).$$

There has also been an improvement of the Hasse-Weil bound.

Theorem 2.2.29 (Serre Bound [18])
Let $F/\mathbb{F}_q$ be a function field of genus g. The number of rational places is bounded by

$$|N - (q+1)| \leq g\lfloor 2q^{1/2} \rfloor.$$

2.3 Coding Theory

This section will give us a short introduction to coding theory. We will focus primarily on cyclic codes, which are a special class of linear codes. First, we introduce block codes and some basic terminology we need throughout this thesis. This section is based on [15] and [19], where one can find a more profound view.

2.3.1 Block Codes

Definition 2.3.1 (Block Code)
An (n, M) block code over a finite alphabet $\mathcal{A}$ is a nonempty subset $\mathcal{C}$ of size M of $\mathcal{A}^n$. We call the parameters n the code length and M the code size.

Throughout this thesis, we will always use a finite field $\mathbb{F}_q$ as an alphabet. If $\mathcal{C}$ is a nonempty subset of $\mathbb{F}_q^n$, we say that $\mathcal{C}$ is a q-ary block code of length n. We call the elements of the code $\mathcal{C}$ code words.
Next, we introduce the Hamming distance, which will define a metric on $\mathbb{F}_q^n$.

Definition 2.3.2 (Hamming Distance)
Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, then the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ is

$$d_H(\mathbf{x}, \mathbf{y}) := |\{i \ : \ 1 \leq i \leq n, \ x_i \neq y_i\}|.$$

It is easy to show that the Hamming distance satisfies the properties of a metric. For any $\mathbf{x}, \mathbf{y}, \mathbf{z}$ in $\mathbb{F}_q^n$, we have

- Positivity: $d_H(\mathbf{x}, \mathbf{y}) \geq 0$, with equality if and only if $\mathbf{x} = \mathbf{y}$.
- Symmetry: $d_H(\mathbf{x}, \mathbf{y}) = d_H(\mathbf{y}, \mathbf{x})$.
- Triangle inequality: $d_H(\mathbf{x}, \mathbf{y}) \leq d_H(\mathbf{x}, \mathbf{z}) + d_H(\mathbf{z}, \mathbf{y})$.

Definition 2.3.3 (Hamming Weight)
Let $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{F}_q^n$, then we call the numbers of nonzero entries of $\mathbf{x}$ the Hamming weight of $\mathbf{x}$, and we write

$$w_H(\mathbf{x}) = |\{i \ : \ x_i \neq 0\}|.$$

Furthermore, we have that $d_H(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} - \mathbf{y})$. Considering block codes, it is of great interest for the code words to have a big Hamming distance between each other.

Definition 2.3.4 (Minimum Distance)
Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a block code. We define the (minimum) distance of $\mathcal{C}$ as

$$d(\mathcal{C}) := min\{d_H(\mathbf{x}, \mathbf{y}) \ : \ \mathbf{x}, \mathbf{y} \in \mathcal{C}, \ \mathbf{x} \neq \mathbf{y}\}.$$

An (n, M) block code with minimum distance d is called an (n, M, d) block code. The minimum distance of a block code is an excellent tool to correct and detect errors. An error occurs if the sent code word doesn't match the received one due to corruption within the channels.

Lemma 2.3.5 (Error Detection, Error Correction [15])
Let $\mathcal{C}$ be a (n, M, d) block code, let $s, t \in \mathbb{N}$.

1. *If $d(\mathcal{C}) \geq s+1$, then up to s errors can be detected.*

2. *If $d(\mathcal{C}) \geq 2t+1$, then up to t errors can be corrected.*

We say that s is the error-detection capacity and t is the error-correction capacity.

Proof.

1. Let $\mathcal{D} : \mathbb{F}_q^n \longrightarrow \mathcal{C} \cup \{'error'\}$ be a decoder defined by

$$\mathcal{D}(\mathbf{y}) = \begin{cases} \mathbf{y}, & \text{if, } \mathbf{y} \in \mathcal{C} \\ 'error', & \text{otherwise .} \end{cases}$$

   If $\mathbf{x} \in \mathcal{C}$ is sent and $\mathbf{y} \in \mathbb{F}_q^n$ with $\mathbf{x} \neq \mathbf{y}$ is received, then $\mathcal{D}$ will fail the error detection if and only if $\mathbf{y}$ is another code word. This can only occur if the number of errors is at least the minimum distance d.

2. Assume again that $\mathbf{x} \in \mathcal{C}$ is sent and $\mathbf{y} \in \mathbb{F}_q^n$ with $\mathbf{x} \neq \mathbf{y}$ is received. Additionally, we assume that $d(x, y) \leq t$ (i.e., at most t errors occurred). Let now $\tilde{\mathbf{x}} \in \mathcal{C}$ such that $d(\tilde{\mathbf{x}}, \mathbf{y}) \leq t$. Then we have

$$d(\mathbf{x}, \tilde{\mathbf{x}}) \leq d(\mathbf{x}, \mathbf{y}) + d(\tilde{\mathbf{x}}, \mathbf{y}) \leq 2t.$$

   This implies that $\mathbf{x} = \tilde{\mathbf{x}}$ and we have a unique code word that is the closest to $\mathbf{y}$.

□

Until now, we considered $\mathcal{C}$ to be some subset of $\mathcal{A}^n$ where $\mathcal{A}$ is an alphabet. Our work will primarily concentrate on linear codes, which are block codes with a specific structure over a finite field.

2.3.2 Linear Codes

Definition 2.3.6 (Linear Code)
An (n, k) linear (block) code over $\mathbb{F}_q$ is a k-dimensional linear subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$.

Using the linearity, we will associate two matrices to each linear code: the generator matrix and the parity-check matrix.

Definition 2.3.7 (Generator Matrix)
A generator matrix of a linear (n, k, d) code is a $k \times n$ matrix whose rows form a basis of the code.

In general, a generator matrix is not unique. However, any generator matrix G of a linear code $\mathcal{C}$ gives us a compact representation of the code. We can write $\mathcal{C} = \{\mathbf{x}G \ : \ \mathbf{x} \in \mathbb{F}_q^k\}$. We will mainly use the letter G to denote a generator matrix.

Definition 2.3.8 (Parity-Check Matrix)
Let $\mathcal{C}$ be a linear (n, k, d) code over $\mathbb{F}_q$. A parity-check matrix H is a $(n-k) \times n$ matrix over $\mathbb{F}_q$ such that for every $\mathbf{c} \in \mathbb{F}_q^n$ we have

$$\mathbf{c} \in \mathcal{C} \iff H\mathbf{c}^T = 0.$$

We will denote any parity-check matrix usually as H. The definition shows that the linear code $\mathcal{C}$ is the (right) kernel of H. In particular, we have the relationship $HG^T = 0$ and we can write $\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_q^n \ : \ H\mathbf{x}^T = 0\}$.
Considering the minimum distance of a linear code, any parity-check matrix of the code characterizes the minimum distance as follows.

Proposition 2.3.9 (Minimum Distance of Linear Code [19])
For a linear code $\mathcal{C}$, the minimum distance equals the minimum weight of a nonzero code word.

$$d(\mathcal{C}) = min\{w_H(\mathbf{c}) \ : \ \mathbf{c} \in \mathcal{C} \setminus \{0\}\}$$

Proof. Since $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$, we have

$$\mathbf{c_1}, \mathbf{c_2} \in \mathcal{C} \implies \mathbf{c_1} - \mathbf{c_2} \in \mathcal{C}.$$

Therefore $d_H(\mathbf{c_1}, \mathbf{c_2}) = d_H(\mathbf{c_1} - \mathbf{c_2}, 0) = w_H(\mathbf{c_1} - \mathbf{c_2})$, and

$$\begin{aligned} d(\mathcal{C}) &= min\{d_H(\mathbf{c_1}, \mathbf{c_2}) \ : \ \mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}, \mathbf{c_1} \neq \mathbf{c_2}\} \\ &= min\{w_H(\mathbf{c_1} - \mathbf{c_2}) \ : \ \mathbf{c_1}, \mathbf{c_2} \in \mathcal{C}, \mathbf{c_1} \neq \mathbf{c_2}\} \\ &= min\{w_H(\mathbf{c}) \ : \ \mathbf{c} \in \mathcal{C} \setminus \{0\}\} \end{aligned} \tag{2.8}$$

□

Theorem 2.3.10 (Characterization of $d(\mathcal{C})$ through H [15])
Let H be a parity-check matrix of a linear code $\mathcal{C} \neq \{0\}$. The minimum distance of $\mathcal{C}$ is the largest integer d such that every set of $d-1$ columns in H is linearly independent.

Proof. Write the parity-check matrix as $H = (\mathbf{h_1} \ \mathbf{h_2} \ \cdots \ \mathbf{h_n})$, where $\mathbf{h_1}, ..., \mathbf{h_n}$ are the columns of H, and let $\mathbf{c} = (c_1, c_2, ..., c_n)$ be a code word of $\mathcal{C}$ with Hamming weight $t > 0$. Let $J \subseteq \{1, 2, .., n\}$ be the support of $\mathbf{c}$, i.e. J is the set of indexes of the t nonzero entries in $\mathbf{c}$. From $H\mathbf{c}^T = 0$ we have

$$\sum_{j \in J} c_j \mathbf{h_j} = \mathbf{0}.$$

Therefore, the t columns in H indexed by J are linearly dependent. Since the smallest possible value of t is the minimum distance d, we must have that all $d-1$ columns of H are linearly independent.
Conversely, every set of t linearly dependent columns in H defines at least one vanishing nontrivial linear combination of the columns of H, with at most t non-zero coefficients in that linear combination. These coefficients form a non-zero code word $\mathbf{c}$ in $\mathcal{C}$ with $w_H(\mathbf{c}) \leq t$.
We know there is no $\mathbf{c}$ in $\mathcal{C}$ with Hamming weight less than the minimum distance d. Therefore, no set of $d-1$ columns in H can be linearly dependent. □

The minimum distance of a code is a good way to measure how many errors can be detected and corrected, as we have seen in Lemma 2.3.5. Another tool that gives us more details about distances within the code is the weight enumerator.

Definition 2.3.11 (Weight Enumerator and Weight Distribution)
Let $\mathcal{C}$ be a linear code of length n and let A_i be the number of code words of Hamming weight i. Then we call

$$A(z) = \sum_{i=0}^{n} A_i z^i$$

the weight enumerator of $\mathcal{C}$. We call the sequence $(A_i)_{i=0}^n$ the weight distribution of $\mathcal{C}$.

Next, we establish the Singleton bound. This gives us an upper bound for the minimum distance of any linear code.

Theorem 2.3.12 (Singleton Bound [19])
For any linear (n, k, d) code $\mathcal{C}$ over $\mathbb{F}_q$,

$$d(\mathcal{C}) \leq n - k + 1$$

Proof. Since the parity-check matrix of $\mathcal{C}$ is a $(n-k) \times n$ matrix, any set of $n-k+1$ columns are linearly dependent. Therefore by Theorem 2.3.10 the minimum distance is at most $n - k + 1$ □

Definition 2.3.13 (Maximum Distance Separable Codes)
A linear (n, k, d) code $\mathcal{C}$ is called maximum distance separable (in short, MDS) if the minimum distance equals the Singleton bound, $d(\mathcal{C}) = n - k + 1$.

2.3.3 Cyclic Codes

A significant class of linear codes are cyclic codes. Throughout this thesis, we will deal a lot with cyclic codes. Therefore, we will establish several important features of them.

Definition 2.3.14 (Cyclic Code)
A linear (n, k, d) code $\mathcal{C}$ over a field $\mathbb{F}_q$ is called cyclic if every cyclic shift of a code word in $\mathcal{C}$ is also a code word. In other words, if $(c_0\ c_1\ ...\ c_{n-1}) \in \mathcal{C}$, then $(c_{n-1}\ c_0\ c_1\ ...\ c_{n-2}) \in \mathcal{C}$.

The most important tool for describing cyclic codes is the following isomorphism between $\mathbb{F}_q^n$ and $\mathbb{F}_q[x]/(x^n - 1)$:

$$(a_0\ a_1\ ...\ a_{n-1}) \in \mathbb{F}_q^n \mapsto a_0 + a_1 x + ... + a_{n-1}x^{n-1} \in \mathbb{F}_q[x]/(x^n - 1).$$

Using this isomorphism, we can describe any code word $c = (c_0\ c_1\ ...\ c_{n-1})$ of any cyclic code $\mathcal{C}$ as a polynomial $c_0 + c_1 x + ... + c_{n-1}x^{n-1}$ in $\mathbb{F}_q[x]/(x^n - 1)$.

Theorem 2.3.15 ([9])
A linear code $\mathcal{C}$ in $\mathbb{F}_q^n$ is cyclic if and only if $\mathcal{C}$ is an ideal in $\mathbb{F}_q[x]/(x^n - 1)$.

Proof.

(i) $\Longrightarrow$: Let $\mathcal{C}$ be cyclic. Then for any code word $c(x) = c_0 + c_1 x + ... + c_{n-1}x^{n-1}$ in $\mathcal{C}$, $xc(x)$ is also in $\mathcal{C}$. It follows that for every $i \in \mathbb{N}$ we have $x^i c(x)$ in $\mathcal{C}$, and since $\mathcal{C}$ is linear $a(x)c(x)$ is in $\mathcal{C}$ for every polynomial $a(x) \in \mathbb{F}_q[x]/(x^n - 1)$. Hence $\mathcal{C}$ is an ideal in $\mathbb{F}_q[x]/(x^n - 1)$.

(ii) $\Longleftarrow$: Let $\mathcal{C}$ be an ideal in $\mathbb{F}_q[x]/(x^n - 1)$. Then for any code word $c(x)$ in $\mathcal{C}$, $xc(x)$ is also in $\mathcal{C}$, i.e. $(c_{n-1}\ c_0\ ...\ c_{n-2}) \in \mathcal{C}$.

□

Notice that $\mathbb{F}_q[x]/(x^n - 1)$ is a principal ideal ring. Therefore, we can describe every element of a cyclic code as a multiple of a nonzero polynomial $g(x)$, which is the monic polynomial of the lowest degree within the ideal. This leads to the following proposition [9].

Proposition 2.3.16 (Generator Polynomial [15])
Let $\mathcal{C}$ be a cyclic (n, k, d) code over $\mathbb{F}_q$ with $k > 0$. Then there is a unique monic polynomial $g(x)$ such that for every $c(x)$ in $\mathbb{F}_q[x]/(x^n - 1)$

$$c(x) \in \mathcal{C} \iff g(x) \mid c(x).$$

We call $g(x)$ the generator polynomial of $\mathcal{C}$.

Proof. Notice that from the requirements on $g(x)$, it follows that if $g(x)$ exists, then since $g(x) \mid g(x)$, we have $g(x) \in \mathcal{C}$. Moreover, since $g(x)$ divides all other monic code words in $\mathcal{C}$, it must be unique.

(i) $\Longleftarrow$: Since by Theorem 2.3.15 $\mathcal{C}$ is the ideal in $\mathbb{F}_q[x]/(x^n - 1)$ generated by $g(x)$, we have that for every $u(x) \in \mathbb{F}_q[x]$, $u(x)g(x) \mod (x^n - 1) \in \mathcal{C}$. Moreover, for every $u(x) \in \mathbb{F}_q[x]$ with $\deg u(x) \leq n - \deg g(x)$, we have $u(x)g(x) \in \mathcal{C}$. Therefore, every code word is a polynomial multiple of $g(x)$ with degree less or equal to n.

(ii) $\Longrightarrow$: Let $c(x) \in \mathcal{C}$ and write $c(x) = u(x)g(x) + r(x)$ with $\deg(r) < \deg(g)$. Since both $c(x)$ and $u(x)g(x)$ are in $\mathcal{C}$, by linearity, we also have $r(x) = u(x)g(x) - c(x) \in \mathcal{C}$. Since $g(x)$ has the smallest degree in $\mathcal{C} \setminus \{0\}$, we get $r(x) = 0$. This implies that $c(x) = u(x)g(x)$, which implies that $g(x) \mid c(x)$.

□

Note that for $k = 0$, we can take $g(x) = x^n - 1$ as the generator polynomial. We have now a new representation of a cyclic code $\mathcal{C}$ by

$$\mathcal{C} = \{u(x)g(x) \ : \ u(x) \in \mathbb{F}_q[x], \ \deg(u) \leq n - \deg(g)\},$$

where $g(x)$ is the generator polynomial of $\mathcal{C}$. From this, we can also conclude that $\deg(g) = n - k$ and we write $u(x) \in \mathbb{F}_q[x]_{\leq k}$. For $g(x) = g_0 + g_1 x + ... + g_{n-k}x^{n-k}$, we obtain the $k \times n$ generator matrix of $\mathcal{C}$

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & & & \\ & g_0 & g_1 & \cdots & g_{n-k} & & 0 \\ & & \ddots & \ddots & \cdots & \ddots & \\ & 0 & & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}.$$

Now, we have a generator matrix of a cyclic code, and therefore, our next step is to define some parity-check matrix H such that $HG^T = 0$. First, we want to see if there is some polynomial $h(x) \in \mathbb{F}_q[x]_{=k}$ such that $h(x)g(x) = 0 \mod (x^n - 1)$ [15]. The following proposition helps us to establish the desired polynomial.

Proposition 2.3.17 ([15])
Let $g(x)$ be the generator polynomial of a cyclic (n, k, d) code $\mathcal{C}$ over $\mathbb{F}_q$. Then

$$g(x) \mid x^n - 1.$$

Proof. Let $x^n - 1 = h(x)g(x) + r(x)$ where $\deg(r) < \deg(g)$. Then

$$r(x) = h(x)g(x) \mod (x^n - 1)$$

and therefore $g(x) \mid r(x)$ in $\mathbb{F}_q[x]/(x^n - 1)$, which implies that $r(x)$ is in $\mathcal{C}$. Since $g(x)$ has the smallest degree of all nonzero polynomials in $\mathcal{C}$ and $\deg(r) < \deg(g)$, we must have $r(x) = 0$. □

Now, we are ready to define the check polynomial and parity check matrix of a cyclic code.

Definition 2.3.18 (Check Polynomial)
Let $\mathcal{C}$ be a cyclic (n, k, d) code with generator polynomial $g(x)$. The check polynomial of $\mathcal{C}$ is the monic polynomial $h(x)$ of degree k obtained by

$$h(x) = \frac{x^n - 1}{g(x)}.$$

Proposition 2.3.19 (Parity-Check Matrix [15])
Let $\mathcal{C}$ be cyclic (n, k, d) code over $\mathbb{F}_q$ with check polynomial $h(x) = h_0 + h_1 x + ... + h_k x^k$. Then the following $(n - k) \times n$ matrix

$$H = \begin{pmatrix} h_k & h_{k-1} & \cdots & h_0 & & \\ & h_k & h_{k-1} & \cdots & h_0 & \\ & & \ddots & \ddots & \cdots & \ddots \\ & 0 & & h_k & h_{k-1} & \cdots & h_0 \end{pmatrix}$$

is a parity-check matrix of $\mathcal{C}$.

Proof. First, observe that $rank(H) = n - k$. Let G be the generator matrix obtained from the generator polynomial $g(x)$ of $\mathcal{C}$; We want to verify that $GH^T = 0$. For every $i = 0, 1, ..., k - 1$ and $l = 0, 1, ..., n - k - 1$, the scalar product of row i in G and row l in H is given by

$$\sum_{j=0}^{n-1} g_{j-i} h_{k+l-j},$$

where $g_j = 0$ for $j \notin \{0, 1, ..., n-k\}$ and $h_j = 0$ for $j \notin \{0, 1, ..., k\}$. Notice, the scalar product of row i in G and row l in H is the coefficient of x^{k+l-i} in the product $g(x)h(x) = x^n - 1$. For the range of values of i and l, we have

$$1 \leq k + l - i \leq n - 1,$$

which implies that the respective coefficients in $g(x)h(x) = x^n - 1$ are all zero. □

One important class of cyclic codes is the class of irreducible cyclic codes. For some generator polynomial $g(x)$ of a cyclic code of length n, we know that $g(x)$ divides $x^n - 1$. Let

$$x^n - 1 = f_1(x)f_2(x) \cdots f_t(x),$$

be the decomposition of $x^n - 1$ into irreducible factors. Then, each generator polynomial of a cyclic code of length n is the product of some of those irreducible factors. Further, if we assume that $gcd(n, q) = 1$, we have that those factors are all different [15].

Definition 2.3.20 (Irreducible Cyclic Code)
Let $x^n - 1 = f_1(x)f_2(x)\cdots f_t(x)$ be a decomposition of $x^n - 1$ into irreducible factors. Then, for all $1 \leq i \leq t$, a cyclic code generated by the generator polynomial $g(x) = (x^n - 1)/f_i(x)$ is called an irreducible cyclic code.

Notice that this definition is equivalent to saying that a cyclic code is irreducible if and only if its check polynomial is nonzero and irreducible.

Example 2.3.21 ([15])
Consider $x^7 - 1$ over $\mathbb{F}_2$. We have the following decomposition of $x^7 - 1$ into irreducible factors,

$$x^7 - 1 = (x - 1)(x^3 + x + 1)(x^3 + x^2 + 1).$$

There are eight cyclic codes of length 7 altogether. One contains only $\mathbf{0}$ as the only code word, and one contains all possible words. By the definition of irreducible cyclic codes, the irreducible cyclic codes of length 7 over $\mathbb{F}_2$ are given by the generator polynomials

$$g_1(x) = (x-1)(x^3+x+1),\ g_2(x) = (x-1)(x^2+x^2+1),\ g_3(x) = (x^3+x+1)(x^3+x^2+1).$$

It is important to mention that in many literature, one can also find the notion of minimal cyclic code instead of irreducible cyclic code. This comes from the fact that the irreducible factors of $x^n - 1$ are maximal ideals in $\mathbb{F}_q[x]/(x^n - 1)$. Therefore, the ideals generated by $(x^n - 1)/f_i(x)$, where $f_i(x)$ is an irreducible factor of $x^n - 1$, are minimal ideals in $\mathbb{F}_q[x]/(x^n - 1)$.
A large class of irreducible cyclic codes are semiprimitive irreducible cyclic codes.

Definition 2.3.22 (Semiprimitive [20])
Let $q = p^l$ and let u be an integer such that $\gcd(p, u) = 1$. We say that p is semiprimitive modulo u if there exists a positive integer j such that $u \mid (p^j + 1)$.

Definition 2.3.23 (Semiprimitive Irreducible Cyclic Code [20])
Let a and u be integers such that $\gcd((q^k - 1)/(q - 1), a) = u$. Let $h_a(x) \in \mathbb{F}_{q^k}[x]$ be the minimal polynomial of $\gamma^{-a} \in \mathbb{F}_{q^k}$. Suppose that $\deg h_a(x) = k$. Then, $h_a(x)$ is the check polynomial of an irreducible cyclic code. Additionally, if $u \geq 2$, and p is semiprimitive modulo u, then we call the code corresponding to $h_a(x)$ semiprimitive irreducible cyclic code.

We also have the following equivalent definition of semiprimitive irreducible cyclic codes, which will appear to be more beneficial for us than the formal definition above.

Definition 2.3.24 (Semiprimitive Irreducible Cyclic Code [20])
Let n, k, and u be integers such that $\gcd(q, n) = 1$, k is the multiplicative order of q modulo n, and $u = \gcd((q^k - 1)/(q - 1), (q^k - 1)/n)$. Then, any (n, k) irreducible cyclic code over $\mathbb{F}_q$ is semiprimitive if $u \geq 2$, and p is semiprimitive modulo u.

We will see that semiprimitive irreducible cyclic codes possess useful properties, which we will utilize to derive their b-symbol weight distribution. Before that, we need to introduce the basic definitions and properties of b-symbol codes.

Chapter 3

b-Symbol Codes

According to traditional information theory, a message is decomposed into individual information units, which may be corrupted while passing through noisy channels. As storage technologies evolve, we've learned that symbols may not always be written and read in a consistent way. For instance, two written symbols may overlap and be read as one information unit. In 2010 and 2011, Cassuto and Blaum introduced a new coding framework with overlapping pairs of symbols as outputs [4] [3]. In 2016, Yaakobi et al. generalized the idea of Cassuto and Blaum of symbol-pair read channels to b-symbol read channels [22]. This Chapter will review those ideas and introduce symbol-pair and b-symbol codes.

3.1 Symbol-Pair Read Channel

In this section, we will go through the ideas of Cassuto and Blaum in [4], where they introduced their symbol-pair read channel model.

Definition 3.1.1 (Symbol-Pair Read Vector)
Let Σ be some alphabet. Let $\mathbf{x} = (x_0, x_1, ..., x_{n-1}) \in \Sigma^n$ be a vector of length n over Σ. Then the symbol-pair read vector of $\mathbf{x}$, denoted by $\pi(\mathbf{x})$, is defined by

$$\pi(\mathbf{x}) = ((x_0, x_1), (x_1, x_2), ..., (x_{n-2}, x_{n-1}), (x_{n-1}, x_0)).$$

If the context is clear, we exclude the brackets of the symbol-pairs and write

$$\pi(\mathbf{x}) = (x_0x_1, x_1x_2, ..., x_{n-1}x_0).$$

Definition 3.1.2 (Pair Distance)
Let $\mathbf{x}$ and $\mathbf{y}$ be in Σ^n. Then, the pair distance is defined by

$$d_2(\mathbf{x}, \mathbf{y}) = d_H((\pi(\mathbf{x}), \pi(\mathbf{y})).$$

Definition 3.1.3 (Pair Weight)
Let $\mathbf{x}$ be in Σ^n. Then the pair weight of $\mathbf{x}$ is defined by

$$w_2(\mathbf{x}) = w_H(\pi(\mathbf{x})).$$

The following proposition shows that we can write the pair distance of $\mathbf{x}$ and $\mathbf{y}$ as the pair weight of their sum.

Proposition 3.1.4 ([22])
For all $\mathbf{x}$ *and* $\mathbf{y}$ *in* Σ^n*, we have*

$$d_2(\mathbf{x}, \mathbf{y}) = w_2(\mathbf{x} - \mathbf{y}).$$

Proof. Let $\mathbf{x}, \mathbf{y} \in \Sigma^n$, then

$$d_2(\mathbf{x}, \mathbf{y}) = d_H(\pi(\mathbf{x}), \pi(\mathbf{y})) = w_H(\pi(\mathbf{x}) - \pi(\mathbf{y})) = w_H(\pi(\mathbf{x} - \mathbf{y})) = w_2(\mathbf{x} - \mathbf{y}).$$

□

Definition 3.1.5 (symbol-pair Code)
For a code $\mathcal{C}$*, we define the symbol-pair code of* $\mathcal{C}$ *by*

$$\pi(\mathcal{C}) = \{\pi(c) \ : \ c \in \mathcal{C}\}.$$

Since we have defined the pair distance and the pair weight for the elements in $\pi(\mathcal{C})$, the next step is to define the minimum distance of a symbol-pair code.

Definition 3.1.6 (Minimum pair distance)
Let $\pi(\mathcal{C})$ *be the symbol-pair code of some code* $\mathcal{C}$*. Then the minimum pair distance of* $\mathcal{C}$ *is the minimum Hamming distance of* $\pi(\mathcal{C})$*, i.e.,*

$$d_2(\mathcal{C}) = d_H(\pi(\mathcal{C})).$$

There is also a connection between the pair distance and the Hamming distance of a code $\mathcal{C}$.

Proposition 3.1.7 ([4])
Let $\mathbf{x}, \mathbf{y} \in \Sigma^n$*, let* $0 < d_H(\mathbf{x}, \mathbf{y}) < n$ *be the Hamming distance between* $\mathbf{x}$ *and* $\mathbf{y}$*. Then we have*

$$d_H(\mathbf{x}, \mathbf{y}) + 1 \leq d_2(\mathbf{x}, \mathbf{y}) \leq 2d_H(\mathbf{x}, \mathbf{y}).$$

In the extreme cases where $d_H(\mathbf{x}, \mathbf{y})$ *is* 0 *or* n*, we have* $d_2(\mathbf{x}, \mathbf{y}) = d_H(\mathbf{x}, \mathbf{y})$.

Proof. Let $\mathbf{x}, \mathbf{y} \in \Sigma^n$, and define $S_H = \{j \ : \ x_j \neq y_j\}$. Clearly if $d_H(\mathbf{x}, \mathbf{y}) = d$, then we also have $|S_H| = d$. Now, let $S_2 = \{i \ : \ (x_i, x_{i+1}) \neq (y_i, y_{i+1})\}$.
Every index $j \in S_H$ appears in exactly two pairs of S_2. This gives us the upper bound $d_2(\mathbf{x}, \mathbf{y}) \leq 2d_H(\mathbf{x}, \mathbf{y})$.
Using the fact that $d_H(\mathbf{x}, \mathbf{y}) < n$, we have that there is at least one pair in S_2 with only one of its indices $i, i+1$ in S_H. This gives us the lower bound $d_H(\mathbf{x}, \mathbf{y}) + 1 \leq d_2(\mathbf{x}, \mathbf{y})$.
If $d_H(\mathbf{x}, \mathbf{y}) = 0$, then $|S_H| = 0$ and therefore also $|S_2| = 0$.
If $d_H(\mathbf{x}, \mathbf{y}) = n$, then $|S_H| = n$ and therefore also $|S_2| = n$. □

We can apply this proposition to the minimum distance of a code.

Corollary 3.1.8
Let $\mathcal{C}$ *be a code of length* n*. If* $0 < d_H(\mathcal{C}) < n$*, then we have*

$$d_H(\mathcal{C}) + 1 \leq d_2(\mathcal{C}) \leq 2d_H(\mathcal{C}).$$

Example 3.1.9 ([22])
Let $\mathcal{C}$ be the single parity-check code of length three.

$$\mathcal{C} = \{(0,0,0),(1,1,0),(1,0,1),(0,1,1)\}$$

Then the symbol-pair code of $\mathcal{C}$ is

$$\pi(\mathcal{C}) = \Big\{\Big(00,00,00\Big),\Big(11,10,01\Big),\Big(10,01,11\Big),\Big(01,11,10\Big)\Big\}$$

The minimum Hamming distance of $\mathcal{C}$ is $d = d_H(\mathcal{C}) = 2$. We have $0 < d < 3$, therefore from Corollary 3.1.8, we must have $3 \leq d_2(\mathcal{C}) \leq 4$. Since each code word has 3 symbol-pairs, we have $d_2(\mathcal{C}) \leq 3$. Therefore, we conclude that $d(\mathcal{C}) = 3$.

We have now established a minimum pair distance of a code $\mathcal{C}$, and therefore, one can ask how many pair errors $\mathcal{C}$ can correct. A pair error occurs when at least one of the two symbols (x_i, x_{i+1}) is changed in $\pi(\mathbf{x})$.

Proposition 3.1.10 ([4])
A code $\mathcal{C}$ can correct t pair-errors if and only if $d_2(\mathcal{C}) \geq 2t+1$.

Proof. Let $d_2(\mathcal{C}) \geq 2t+1$. Assume that $\pi(\mathbf{x}) \in \pi(\mathcal{C})$ is sent and $\pi(\mathbf{y}) \neq \pi(\mathbf{x}) \in (\Sigma \times \Sigma)^n$ is received.
Assume that $d_2(\mathbf{x},\mathbf{y}) \leq t$. Let $\tilde{\mathbf{x}} \in \mathcal{C}$ such that $d_2(\tilde{\mathbf{x}},\mathbf{y}) \leq t$. Then we have

$$d_2(\mathbf{x},\tilde{\mathbf{x}}) \leq d_2(\mathbf{x},\mathbf{y}) + d_2(\tilde{\mathbf{x}},\mathbf{y}) \leq 2t.$$

Since $d_2(\mathcal{C}) \geq 2t+1$ we have $\mathbf{x} = \tilde{\mathbf{x}}$. This implies that $\mathbf{x}$ is the unique code word with the smallest pair distance to $\mathbf{y}$. Conversely, we assume that $d = d_2(\mathcal{C}) < 2t+1$. Suppose that $\mathbf{x},\tilde{\mathbf{x}} \in \mathcal{C}$ are such that $d_2(\mathbf{x},\tilde{\mathbf{x}}) = d$. Again, let's assume $\pi(\mathbf{x})$ is sent and $\pi(\mathbf{y})$ is received. Further, suppose that $\pi(\mathbf{y})$ is such that if d is even $\pi(\mathbf{y})$ has $d/2$ of the pair symbols of $\pi(\tilde{\mathbf{x}})$ where $\pi(\tilde{\mathbf{x}})$ differs from $\pi(\mathbf{x})$ and the remaining symbols are the same as in $\pi(\mathbf{x})$. If d is odd $\pi(\mathbf{y})$ has $(d+1)/2$ of the pair symbols of $\pi(\tilde{\mathbf{x}})$ where $\pi(\tilde{\mathbf{x}})$ differs from $\pi(\mathbf{x})$ and the remaining symbols are the same as in $\pi(\mathbf{x})$. Since we constructed $\pi(\mathbf{y})$ this way, it has at most t pair errors. we have

$$d_2(\mathbf{x},\mathbf{y}) = d_2(\tilde{\mathbf{x}},\mathbf{y}) = \frac{d}{2} \text{ if } d \text{ is even},$$

$$d_2(\mathbf{x},\mathbf{y}) = d_2(\tilde{\mathbf{x}},\mathbf{y}) = \frac{d+1}{2} \text{ if } d \text{ is odd}\ .$$

Therefore, there is no unique symbol-pair code word that is closest to $\pi(\mathbf{y})$. Therefore, $\mathcal{C}$ cannot correct t pair errors. We conclude that $d_2(\mathcal{C}) \geq 2t+1$. □

The proof of Proposition 3.1.10 is very similar to the proof in the Hamming distance case. Considering the error correction capacity of symbol-pair codes, it is of great interest that $d_2(\mathcal{C}) = 2d_H(\mathcal{C})$ as in Corollary 3.1.8. The following example shows a way to achieve that.

Example 3.1.11 (Interleaving Code Words of Two Codes [4])
We know from Corollary 3.1.8 that there is the possibility for $d_2(\mathcal{C}) = 2d_H(\mathcal{C})$. One way to

obtain this is by interleaving two code words.
Let $\mathcal{C}_1$ be an (n, m_1, d) code, and $\mathcal{C}_2$ an (n, m_2, d) code. Then we define $\mathcal{C}$ as

$$\mathcal{C} = \{(x_0, y_0, x_1, y_1, ..., x_{n-1}, y_{n-1}) \ : \ \mathbf{x} \in \mathcal{C}_1, \mathbf{y} \in \mathcal{C}_2\}.$$

$\mathcal{C}$ is a $(2n, m_1 m_2, d)$ code. Moreover, we have that $d_2(\mathcal{C}) = 2d_H(\mathcal{C})$.

Proof. Let $S_H = \{j \ : \ x_j \neq y_j\} = d_H(\mathbf{x}, \mathbf{y})$ and let $S_H = \bigcup_{l=1}^{L} B_l$ be a minimal partition of the set S_H into subsets of consecutive indices. This means that each B_l consists of consecutive integers of S_H. The requirement that the partition has to be minimal guarantees that every two adjacent indices $i, i+1$ are in the same B_l. Suppose B_l consists of all consecutive integers between s_l and e_l including s_l and e_l. Then, one can determine $d_2(\mathbf{x}, \mathbf{y})$ by taking the sum of the sizes of all pair subsets

$$\{(s_l - 1, s_l), (s_l, s_l + 1), ..., (e_l, e_l + 1)\}.$$

The number of pairs in these subsets is given by $|B_l| + 1$. This implies that

$$D_2(\mathbf{x}, \mathbf{y}) = D_H(\mathbf{x}, \mathbf{y}) + L.$$

Note that for the interleaved code $\mathcal{C}$, the Hamming distance on $\mathcal{C}_1$'s or $\mathcal{C}_2$'s coordinates (or both) is at least d. Further, for $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ none of the coordinates of the same $\mathcal{C}_i$, for $i = 1, 2$, where $\mathbf{x}$ differs from $\mathbf{y}$ are in consecutive locations. Therefore, the minimal partition of S_H is at least d. This implies that $d_2(\mathcal{C}) \geq 2d$. Using Corollary 3.1.8, we conclude that $d_2(\mathcal{C}) = 2d_H(\mathcal{C})$. □

Example 3.1.11 shows that symbol-pair codes constructed by interleaving two codes of the same Hamming distance give the best possible pair distance for the given Hamming distance. However, these codes are generally inferior to directly constructed codes. This is also true for the case where $\mathcal{C}_1$ and $\mathcal{C}_2$ are themselves optimal codes with respect to their Hamming metric. This is shown in the following example.

Example 3.1.12 ([4])
Consider the $(30, 22)$ shortened cyclic code over $\mathbb{F}_2$ with generator polynomial $1+x^2+x^3+x^8$. This code has minimum pair distance $d_2 = 7$, and can therefore correct $\lfloor \frac{7-1}{2} \rfloor = 3$ pair errors.
Now, we take the perfect $(15, 11)$ binary Hamming code, and by interleaving it with itself, we obtain an $(30, 22)$ code with minimum pair distance $d_2 = 6$, which can correct $\lfloor \frac{6-1}{2} \rfloor = 2$ pair errors.

Interleaving two codes indeed gives the maximum pair distance for a given minimum Hamming distance, but this minimum Hamming distance is, in general, bad for the length of the code. Therefore, it is a better approach to try directly to construct symbol-pair codes with good minimum pair distance. We will focus more on cyclic codes, which provide large minimum pair distances.

3.2 Cyclic Codes over Symbol-Pair Read Channels

This section aims to provide a lower bound on the minimum pair distance of cyclic codes. This will show that symbol-pair codes, constructed from cyclic codes, have a good pair distance. We will also look at an example of one decoding algorithm. This will give us a view of the challenges of decoding symbol-pair codes.

3.2.1 Pair Distance of Cyclic Codes

In this part, we will consider only cyclic codes over $\mathbb{F}_2$ for our convenience. Therefore, we can think of non-zero entries as ones. The following lemma will help us provide a lower bound for $d_2(\mathcal{C})$.

Lemma 3.2.1 ([22])
For any $\mathbf{x} = (x_0, ..., x_{n-1}) \in \mathbb{F}_2^n$, *let* $\mathbf{x}' = (x_0 + x_1, x_1 + x_2, ..., x_{n-1} + x_0)$, *then*

$$w_2(\mathbf{x}) = w_H(\mathbf{x}) + \frac{w_H(\mathbf{x}')}{2}.$$

Proof. Let

$$S_0 = \{i \ : \ (x_i, x_{i+1}) \neq (0,0) \ \text{ and } x_i = 1\},$$
$$S_1 = \{i \ : \ (x_i, x_{i+1}) \neq (0,0) \ \text{ and } x_i = 0\}.$$

Notice that $|S_0| = d_H(\mathbf{x})$, $S_0 \cap S_1 = \emptyset$, and $w_2(\mathbf{x}) = |S_0| + |S_1|$. We have $i \in S_1$ if and only if $x_i = 0$ and $x_1 = 1$ for all $0 \leq i \leq n-1$. In this case, we have $x'_i = x_i + x_{i+1} = 1$. This implies that

$$|S_1| = |\{i \ : \ x_i = 0 \ \text{ and } x'_i = 1\}|.$$

Additionally, for every $\mathbf{x} \in \mathbb{F}_2^n$, we have

$$|\{i \ : \ x_i = 0 \ \text{ and } x'_i = 1\}| = |\{i \ : \ x_i = 1 \ \text{ and } x'_i = 1\}|.$$

The sum of these two cardinalities equals $w_H(\mathbf{x}')$. We obtain

$$|S_1| = \frac{w_H(\mathbf{x}')}{2}$$

and hence

$$w_2(\mathbf{x}) = |S_0| + |S_1| = w_H(\mathbf{x}) + \frac{w_H(\mathbf{x}')}{2}.$$

□

With this lemma, we can establish a lower bound on the minimum pair distance of linear cyclic codes.

Theorem 3.2.2 ([22])
Let $\mathcal{C}$ *be a linear cyclic code of dimension greater than one. Then*

$$d_2(\mathcal{C}) \geq d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil.$$

Proof. Let $\mathbf{x} = (x_0, ..., x_{n-1}) \in \mathcal{C} \setminus \{\mathbf{0}\}$. First, we assume that $\mathbf{x} \neq \mathbf{1}$. Since the code is cyclic, we have that $(x_1, ..., x_{n-1}, x_0) \in \mathcal{C}$, and therefore

$$\mathbf{x}' = (x_0, ..., x_{n-1}) + (x_1, ..., x_{n-1}, x_0) \in \mathcal{C}.$$

Note that $w_H(\mathbf{x}')$ is even and from $\mathbf{x} \neq 1$, we have $\mathbf{x}' \neq 0$. Hence $w_H(\mathbf{x}') \geq 2\lceil d_H(\mathcal{C})/2 \rceil$. Since $w_H(\mathbf{x}) \geq d_H(\mathcal{C})$, we have that

$$w_2(\mathbf{x}) = w_H(\mathbf{x}) + \frac{w_H(\mathbf{x}')}{2} \geq d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil.$$

Now we have to consider the $\mathbf{x} = \mathbf{1} \in \mathcal{C}$ case. Then $w_2(\mathbf{x}) = n$. If we show that $d_H(\mathcal{C}) \leq \lfloor 2n/3 \rfloor$, then we have

$$d_H(\mathcal{C}) + \left\lceil \frac{d_H(\mathcal{C})}{2} \right\rceil \leq \lfloor 2n/3 \rfloor + \lceil n/3 \rceil = n = w_2(\mathbf{x})$$

and we are done. By assumption, the dimension of $\mathcal{C}$ is greater than one, and therefore, the code contains at least three distinct non-zero code words. We choose two of them and call them $\mathbf{x_1}$ and $\mathbf{x_2}$. Let's assume that $d_H(\mathcal{C}) > \lfloor 2n/3 \rfloor$. Then $w_H(\mathbf{x_1}), w_H(\mathbf{x_2}) \geq \lfloor 2n/3 \rfloor + 1$, which implies that

$$w_H(\mathbf{x_1} + \mathbf{x_2}) \leq 2n - 2 \cdot (\lfloor 2n/3 \rfloor + 1) \leq \lfloor 2n/3 \rfloor,$$

which contradicts the assumption. □

Theorem 3.2.2 shows that linear cyclic codes have large minimum pair distances. This allows us to correct a large amount of symbol-pair errors. We will dedicate the next part of this section to decoding symbol-pair codes over binary cyclic codes. The challenge in finding a decoding algorithm for symbol-pair codes is dealing with multiple types of errors. For instance, the symbol-pair (x_{i-1}, x_i) in $\pi(\mathbf{x})$ can have an error in the first, second, or both symbols.

3.2.2 Decoding Symbol-Pair Codes over Cyclic Codes

In this section, we look at a decoder, introduced in [22], and some examples.
Let $\mathcal{C}$ be a linear cyclic code with minimum distance $d_H(\mathcal{C}) = 2t + 1$. Furthermore, we assume that $\mathcal{C}$ is a binary code, meaning a code over $\mathbb{F}_2$. Suppose a decoder exists for $\mathcal{C}$ that can correct up to t errors. The goal is to use this decoder to construct a decoder for the code $\pi(\mathcal{C})$ which corrects up to $t_0 = \lfloor \frac{3t+1}{2} \rfloor$ symbol-pair errors.
Suppose the dimension of $\mathcal{C}$ is greater than one. As stated in the proof of Theorem 3.2.2, this implies that $d_H(\mathcal{C}) \leq \lfloor 2n/3 \rfloor$. This leads to $d_H(\mathcal{C}) = 2t + 1 \leq \lfloor 2n/3 \rfloor$, and therefore we obtain the inequalities

$$t \leq \left\lfloor \frac{2n/3 - 1}{2} \right\rfloor < n/3,$$
$$t_0 < n/2.$$

Let $D_{\mathcal{C}} : \Sigma^n \to \mathcal{C} \cup \{F\}$ be the decoder of code $\mathcal{C}$, where F denotes a failure in the decoding process. For instance, if $\mathbf{y} \in \Sigma^n$ is received, then $D_{\mathcal{C}}(\mathbf{y}) = \hat{\mathbf{c}} \in \mathcal{C} \cup \{F\}$. Suppose $\mathbf{c} \in \mathcal{C}$ is the transmitted code word; then we have two possible cases.

Case 1: $d_H(\mathbf{c}, \mathbf{y}) \leq t$
In this case, we must have $\hat{\mathbf{c}} = \mathbf{c}$.

Case 2: $d_H(\mathbf{c}, \mathbf{y}) > t$
In this case, we either have that $\hat{\mathbf{c}}$ is not equal to $\mathbf{c}$, but still a code word with Hamming distance $d_H(\hat{\mathbf{c}}, \mathbf{y}) \leq t$, or $\hat{\mathbf{c}} = F$, which indicates that $\hat{\mathbf{c}} \notin \mathcal{C}$.

For the design of the symbol-pair decoder, we have to introduce the double-repetition code of $\mathcal{C}$.

Definition 3.2.3 (Double-Repetition Code)
Let $\mathcal{C}$ be a code. Then the double-repetition code of $\mathcal{C}$ is defined by

$$\mathcal{C}_2 = \{(\mathbf{c}, \mathbf{c}) \ : \ \mathbf{c} \in \mathcal{C}\}.$$

Further, if $\mathcal{C}$ is a code of length n, then $\mathcal{C}_2$ is a code of length $2n$ with minimum Hamming distance $d_H(\mathcal{C}_2) = 2d_H(\mathcal{C})$. Therefore, the double-repetition code $\mathcal{C}_2$ can correct up to $2t$ errors. Suppose that there exists a decoder for $\mathcal{C}_2$ denoted by $D_{\mathcal{C}_2} : \Sigma^n \times \Sigma^n \to \mathcal{C}_2 \cup \{F\}$. For simplicity, we assume that for any received word $(\mathbf{y_1}, \mathbf{y_2}) \in \Sigma^n \times \Sigma^n$ with $d_H((\mathbf{y_1}, \mathbf{y_2}), (\hat{\mathbf{c}}, \hat{\mathbf{c}})) \leq 2t$, for some $\hat{\mathbf{c}} \in \mathcal{C}$, the decoder $D_{\mathcal{C}_2}$ returns $D_{\mathcal{C}_2}(\mathbf{y_1}, \mathbf{y_2}) = \hat{\mathbf{c}}$.
Let $\mathbf{c} \in \mathcal{C}$ and $\pi(\mathbf{c}) \in \pi(\mathcal{C})$ be its corresponding symbol-pair code word. Our goal is to describe a decoder $D_\pi : (\Sigma \times \Sigma)^n \to \pi(\mathcal{C}) \cup \{F\}$, that can correct the error $\mathbf{e} \in (\Sigma \times \Sigma)^n$ of a received word $\mathbf{y} = \pi(\mathbf{c}) + \mathbf{e}$, if the Hamming weight of $\mathbf{e}$ satisfies

$$\mathbf{e} \leq t_0 = \left\lfloor \frac{3t+1}{2} \right\rfloor.$$

Suppose the received word has the form

$$\mathbf{y} = \pi(\mathbf{c}) + \mathbf{e} = ((y_{0,0}, y_{0,1}), (y_{1,0}, y_{1,1}), (y_{2,0}, y_{2,1}), ..., (y_{n-1,0}, y_{n-1,1})).$$

We introduce the three vectors related to $\mathbf{y}$:

$$\begin{aligned}
\mathbf{y_L} &= (y_{0,0}, y_{1,0}, ..., y_{n-1,0}), \\
\mathbf{y_R} &= (y_{0,1}, y_{1,1}, ..., y_{n-1,1}), \\
\mathbf{y_S} &= \mathbf{y_L} + \mathbf{y_R} \\
&= (y_{0,0} + y_{0,1}, y_{1,0} + y_{1,1}, ..., y_{n-1,0} + y_{n-1.1}).
\end{aligned}$$

We know the vector $\mathbf{y}$ has at most t_0 symbol-pair errors. Thus, the vectors $\mathbf{y_L}$ and $\mathbf{y}_R$ also have at most t_0 symbol-pair errors.
One can consider $\mathbf{y}_L$ as a noisy version of $\mathbf{c} = (c_0, c_1, ..., c_{n-1})$, and $\mathbf{y}_R$ as a noisy version of the left-cycle shift of $\mathbf{c}$, that is $(c_1, c_2, ..., c_{n-1}, c_0)$. Therefore, the right cycle shift of $\mathbf{y}_R$, which we denote by $\mathbf{y}_R^{(1)} = (y_{n-1,1}, y_{0,1}, y_{1,1}, ..., y_{n-2,1})$, and $\mathbf{y_L}$ can both be viewed as a noisy version of $\mathbf{c}$.
Considering the vector $\mathbf{y_S}$, $\mathbf{y_S}$ has at most t_0 errors with respect to the code word $\mathbf{c}' = (c_0 + c_1, c_1 + c_2, ..., c_{n-1} + c_0)$. We will show that one can derive the value of $\mathbf{c}$ from $\mathbf{c}'$. This is not true in general, but in this setting, it works [22].

Lemma 3.2.4 ([22])
In the symbol-pair read channel setting described above, if the code word $\mathbf{c}' \in \mathcal{C}$ is successfully recovered, then we can uniquely determine the code word $\mathbf{c}$.

Proof. Assume that $\mathbf{c}'$ is decoded successfully, and we know the value of

$$\mathbf{c}' = (c'_0, c'_1, ..., c'_{n-1}) = (c_0 + c_1, c_1 + c_2, ..., c_{n-1} + c_0).$$

Notice that $c_0 + (c_0 + c_1) + (c_1 + c_2) + \cdots + (c_{i-1} + c_i) = c_i$. I.e., $\mathbf{c}$ satisfies

$$c_i = c_0 + \sum_{j=0}^{i-1} c'_j.$$

Hence if we define $\tilde{\mathbf{c}} = (\tilde{c}_0, ..., \tilde{c}_{n-1})$ by $\tilde{c}_0 = 0$ and for $1 \leq i \leq n-1$, $\tilde{c}_i = \sum_{j=0}^{i-1} c'_j$, then the code word $\mathbf{c}$ is either $\tilde{\mathbf{c}}$ or $\tilde{\mathbf{c}} + \mathbf{1}$, depending on the value of c_0. The Hamming distance between $\mathbf{y_L}$ and $\mathbf{c}$ is at most t_0 and $d_H(\mathbf{c}, \mathbf{c}+\mathbf{1}) = n$. Since $t_0 < n/2$, we have

$$d_H(\mathbf{y_L}, \mathbf{c}+\mathbf{1}) = n - t_0 > t_0.$$

Hence, if $d_H(\mathbf{y_L}, \tilde{\mathbf{c}}) < d_H(\mathbf{y_L}, \tilde{\mathbf{c}}+\mathbf{1})$, then $\mathbf{c} = \tilde{\mathbf{c}}$, otherwise, $\mathbf{c} = \tilde{\mathbf{c}} + \mathbf{1}$. □

We will denote $\mathbf{c}'^*$ as the code word $\mathbf{c}$ obtained from $\mathbf{c}'$ as in Lemma 3.2.4.
The received vector $\mathbf{y}$ has at most t_0 symbol-pair errors. The symbol-pair errors consist of either one symbol being wrong or both. In other words, each symbol-pair error is either a one-bit or two-bit error. We define

$$E_1 = \text{"number of single-bit errors"},$$
$$E_2 = \text{"number of double-bit errors"}.$$

We have $E_1 + E_2 \leq t_0$. The number of errors in $\mathbf{y_S}$ is E_1, and the number of errors in $(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ is $E_1 + 2E_2$. The next lemma is significant for validating the symbol-pair error decoding algorithm.

Lemma 3.2.5 ([22])
If $\mathbf{c} \in \mathcal{C}$, $\mathbf{y} = \pi(\mathbf{c}) + \mathbf{e}$, and $w_H(\mathbf{e}) \leq t_0$, then either $D_{\mathcal{C}}(\mathbf{y_S}) = \mathbf{c}'$ or $D_{\mathcal{C}_2}((\mathbf{y_L}, \mathbf{y_R^{(1)}})) = \mathbf{c}$.

Proof. If $E_1 \leq t$ then the decoder $D_{\mathcal{C}}(\mathbf{y_S})$ is successful, and we obtain $D_{\mathcal{C}}(\mathbf{y_S}) = \mathbf{c}'$.
If $E_1 \geq t+1$, then $E_2 \leq t_0 - (t+1)$. Thus, the number of errors in $(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ satisfies

$$E_1 + 2E_2 \leq t_0 + t_0 - (t+1) = 2\left\lfloor \frac{3t+1}{2} \right\rfloor - (t+1) \leq 2t.$$

Therefore, the decoder $D_{\mathcal{C}_2}(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ is successful and we obtain $D_{\mathcal{C}_2}(\mathbf{y_L}, \mathbf{y_R^{(1)}}) = \mathbf{c}$ □

Lemma 3.2.5 implies that at least one of the two decoders $D_{\mathcal{C}}$ and $D_{\mathcal{C}_2}$ succeeds. So the goal of the decoder D_π is to find out which of the two decoders is the successful one. We will define the step-by-step process of the decoder D_π and eventually prove that this decoder works.

Definition 3.2.6 (Decoder D_π [22])
Given a received vector $\mathbf{y}$, the decoder output $D_\pi(\mathbf{y}) = \hat{\mathbf{c}} = \mathbf{c}$ is calculated as follows.

Step 1: $\mathbf{c_1} = D_{\mathcal{C}}(\mathbf{y_S})$, $e_1 = d_H(\mathbf{c_1}, \mathbf{y_S})$.

Step 2: $\mathbf{c_2} = D_{\mathcal{C}_2}\big((\mathbf{y_L}, \mathbf{y_R^{(1)}})\big)$, $e_2 = d_H((\mathbf{c_2}, \mathbf{c_2}), (\mathbf{y_L}, \mathbf{y_R^{(1)}}))$.

Step 3: *If* $\mathbf{c_1} = F$ *or* $w_H(\mathbf{c_1})$ *is odd then* $\hat{\mathbf{c}} = \mathbf{c_2}$.

Step 4: *If* $e_1 \leq \lfloor \frac{t+2}{2} \rfloor$, *then* $\hat{\mathbf{c}} = \mathbf{c_1^*}$.

Step 5: *If* $e_1 > \lfloor \frac{t+2}{2} \rfloor$, *let* $e_1 = \lfloor \frac{t+2}{2} \rfloor + a$, *where* $1 \leq a \leq \lceil \frac{t}{2} \rceil - 1$.

a) If $e_2 \leq t_0 + a$, *then* $\hat{\mathbf{c}} = \mathbf{c_2}$.

b) Otherwise, $\hat{\mathbf{c}} = \mathbf{c_1^*}$.

Theorem 3.2.7 ([22])
The output of the decoder D_π satisfies $D_\pi(\mathbf{y}) = \hat{\mathbf{c}} = \mathbf{c}$.

Proof. We know from Lemma 3.2.5 that at least one of the two decoders in steps 1 and 2 succeeds. The purpose of steps 3-5 is to determine which decoder is the successful one.

Step 3: Recall that $\mathbf{y_S}$ is a noisy version of the code word $\mathbf{c}'$. The decoding in step 1 attempts to decode $\mathbf{c}'$, which has even Hamming weight. If either $\mathbf{c_1} = F$ or $w_H(c_1)$ is odd, then we know that the decoding operation in step 1 has failed. This implies that the decoding operation in step 2 must be the successful one. This also means that in steps 4 and 5, $w_H(\mathbf{c_1})$ must be even.

Step 4: We want to show that if $e_1 \leq \lfloor \frac{t+2}{2} \rfloor$, then $E_1 \leq \lfloor \frac{t+2}{2} \rfloor$ as well, which means that the decoder in step 1 succeeded.
Notice that the minimum Hamming distance of the code $\mathcal{C}$ is $2t+1$. If $\mathbf{c_1}$ is not the initial sent code word and is wrong due to a miscorrection in step 1, then the number E_1 of actual errors in $\mathbf{y_S}$ has to be bigger than the minimum Hamming distance of $\mathcal{C}$ minus the maximum possible weight of the error vector e_1. I.e.,

$$E_1 \geq 2t + 2 - \left\lfloor \frac{t+2}{2} \right\rfloor = \left\lceil \frac{3t}{2} \right\rceil + 1 > t_0.$$

This contradicts the fact that the number of errors in $\mathbf{y_S}$ is at most t_0. This implies that the decoding operation $\mathbf{c_1} = D_{\mathcal{C}}(\mathbf{y_S}) = \mathbf{c}'$ succeeds. From Lemma 3.2.4, we conclude

$$\hat{\mathbf{c}} = \mathbf{c_1^*} = \mathbf{c}'^* = \mathbf{c}.$$

Step 5: Now, in step 5 we know that $w_H(\mathbf{c_1})$ is even and $e_1 > \lfloor \frac{t+2}{2} \rfloor$. Since $e_1 \leq t$, we can write $e_1 = \lfloor \frac{t+2}{2} \rfloor + a$, where $1 \leq a \leq \lceil \frac{t}{2} \rceil - 1$.
Assume the decoding in step 2 has failed. Therefore, by Lemma 3.2.5, we know that the decoding in step 1 succeeds. This implies that

$$E_1 = e_1 = \left\lfloor \frac{t+2}{2} \right\rfloor + a.$$

From $E_1 + E_2 \leq t_0$ we obtain the inequality

$$E_2 \leq t_0 - E_1 = t_0 - \left\lfloor \frac{t+2}{2} \right\rfloor - a = \left\lfloor \frac{3t+1}{2} \right\rfloor - \left\lfloor \frac{t+2}{2} \right\rfloor - a \leq t - a.$$

The total number of errors in $(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ is

$$E_1 + 2E_2 \leq t_0 + t - a = \left\lfloor \frac{5t+1}{2} \right\rfloor - a.$$

By assumption, the decoder $D_{\mathcal{C}_2}(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ fails, and therefore the weight of the error vector in step 2, e_2, must satisfy

$$e_2 \geq d_H(\mathcal{C}_2) - (E_1 + 2E_2) = 2(2t+1) - (E_1 + 2E_2) \geq 4t + 2 - \left(\left\lfloor \frac{5t+1}{2} \right\rfloor - a \right)$$
$$\geq \left\lfloor \frac{3t+1}{2} \right\rfloor + a + 1 = t_0 + a + 1.$$

Hence, for $e_2 \leq t_0 + a$, e_2 doesn't satisfy the inequality, and therefore the decoder in step 2 succeeds if $e_2 \leq t_0 + a$ with $\hat{\mathbf{c}} = \mathbf{c_2}$.
On the other hand, we can assume that the decoding in step 1 fails. Analog to step 4, this means that the number of errors E_1 in $\mathbf{y_S}$ satisfies

$$E_1 \geq 2t + 2 - \left(\left\lfloor \frac{t+2}{2} \right\rfloor + a \right) = \left\lceil \frac{3t}{2} \right\rceil - (a-1) = t_0 + a - 1.$$

Since $E_1 + E_2 \leq t_0$, it follows that E_2 satisfies $0 \leq E_2 \leq a - 1$, and $E_1 + 2E_2$, the total number of errors in $(\mathbf{y_L}, \mathbf{y_R^{(1)}})$, satisfies

$$t_0 - (a-1) \leq E_1 + 2E_2 = (E_1 + E_2) + E_2 \leq t_0 + a - 1.$$

Since the decoding operation $D_{\mathcal{C}_2}(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ succeeds, and

$$t_0 - (a-1) \leq e_2 \leq t_0 + a - 1,$$

the decoder in step 1 succeeds if $e_2 > t_0 + a$.

□

The decoder D_π works, and we will look at some examples.

Example 3.2.8 ([22])
Let $\mathcal{C}$ be the cyclic binary triple-error correcting BCH code of length 15. We have $d_H(\mathcal{C}) = 7$, $d_2(\mathcal{C}) = 7 + \lceil \frac{7}{2} \rceil = 11$, $t = 3$, and $t_0 = \lfloor \frac{3t+1}{2} \rfloor = 5$. This implies that the code can correct up to 5 symbol-pair errors. Assume $\mathbf{0}$ was sent and $\mathbf{y}$ received, where

$$\mathbf{y} = (00, 11, 10, 00, 00, 00, 11, 00, 10, 00, 00, 11, 00, 00, 00).$$

We obtain the vectors

$$\begin{aligned}
\mathbf{y_L} &= (0,1,1,0,0,0,1,0,1,0,0,1,0,0,0), \\
\mathbf{y_R} &= (0,1,0,0,0,0,1,0,0,0,0,1,0,0,0), \\
\mathbf{y_R^{(1)}} &= (0,0,1,0,0,0,0,1,0,0,0,0,1,0,0), \\
\mathbf{y_S} &= (0,0,1,0,0,0,0,0,1,0,0,0,0,0,0).
\end{aligned}$$

Now, we go through all the steps in the decoding algorithm of D_π and show that the initial code word is $\mathbf{0}$.

Step 1: We calculate $\mathbf{c_1} = D_{\mathcal{C}}(\mathbf{y_S})$, $e_1 = d_H(\mathbf{c_1}, \mathbf{y_S})$. Since we have two errors in $\mathbf{y_S}$ and $D_{\mathcal{C}}$ can correct up to three errors, we obtain $\mathbf{c_1} = \mathbf{0}$ and $e_1 = 2$.

*Step 2: We calculate $\mathbf{c_2} = D_{\mathcal{C}_2}((\mathbf{y_L}, \mathbf{y_R^{(1)}}))$, $e_2 = d_H((\mathbf{c_2}, \mathbf{c_2}), (\mathbf{y_L}, \mathbf{y_R^{(1)}}))$.
The vector $(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ suffered 8 errors. The code $\mathcal{C}_2$ has minimum distance 14, and therefore, the decoder $D_{\mathcal{C}_2}$ can correct up to 6 errors. Hence, the output $\mathbf{c_2}$ is either a failure F, meaning that there is no code word with Hamming distance at most 6 from $(\mathbf{y_L}, \mathbf{y_R^{(1)}})$, or some code word $\mathbf{c_2}$ such that $e_2 = 6$.*

Step 3: We have $\mathbf{c_1} = \mathbf{0}$ *and* $w_H(\mathbf{c_1}) = 0$*. Therefore, in step 3, we cannot determine which decoder is correct, so we follow the next step.*

Step 4: We have $2 \leq \lfloor \frac{3+2}{2} \rfloor = 2$*. This implies that the decoder in step 1 was successful, and* $\hat{\mathbf{c}} = \mathbf{0}^* = \mathbf{0}$*. Note that* $\mathbf{0}^*$ *is either* $\mathbf{0}$ *or* $\mathbf{1}$*, but since the Hamming distance from* $\mathbf{1}$ *to the received word is too large, we can ignore this case.*

As a second example, let us assume that the received vector is

$$\mathbf{y} = (10, 00, 01, 00, 01, 00, 00, 00, 00, 10, 00, 00, 00, 10, 00).$$

Again, we can deduce the related vectors

$$\begin{aligned}
\mathbf{y_L} &= (1,0,0,0,0,0,0,0,0,1,0,0,0,1,0),\\
\mathbf{y_R} &= (0,0,1,0,1,0,0,0,0,0,0,0,0,0,0),\\
\mathbf{y_R^{(1)}} &= (0,0,0,1,0,1,0,0,0,0,0,0,0,0,0),\\
\mathbf{y_S} &= (1,0,1,0,1,0,0,0,0,1,0,0,0,1,0).
\end{aligned}$$

Step 1: $\mathbf{y_S}$ *has 5 errors. Therefore, either we have* $\mathbf{c_1} = F$ *or* $\mathbf{c_1}$ *is a code word of weight seven or eight with a distance of two or three from* $\mathbf{y_S}$*.*

Step 2: The decoder $D_{\mathcal{C}_2}(\mathbf{y_L}, \mathbf{y_R^{(1)}})$ *succeeds and we obtain* $\mathbf{c_2} = \mathbf{0}$ *and* $e_2 = 5$*.*

Step 3: We can't determine the exact value of $w_H(\mathbf{c_1})$*. Thus, we go to the next step.*

Step 4: Since e_1 *is either* 2 *or* 3*, we can't assure that* $e_1 \leq \lfloor \frac{5}{2} \rfloor = 2$ *is satisfied. We need step 5 to determine which decoder was successful.*

Step 5: Notice that for $a = 1$ *we have* $e_2 = 5 \leq t_0 + a = 6$*. Hence, the condition in step* 5*a) holds, and we can conclude that the decoder in step 2 is the successful one, which gives us* $\hat{\mathbf{c}} = \mathbf{0}$*.*

Notice that the only assumption we made on the decoders $D_{\mathcal{C}}$ and $D_{\mathcal{C}_2}$ was on the error correction capability. Therefore, the decoder D_π performs as well as the best existing decoders for cyclic binary codes. In general, the challenge in finding decoders for symbol-pair codes and b-symbol codes is that there are more types of occurring errors. We've seen that when we divided the error types into single-bit and double-bit errors. There is still much work to be done to find good decoders for general symbol-pair codes or b-symbol codes over cyclic codes.
Since we already mentioned b-symbol codes, we can now introduce this concept, which is a generalization of symbol-pair codes.

3.3 b-Symbol Read Channel

In this section, we want to generalize the ideas from symbol-pair read channels to b-symbol read channels, where the entries of a b-symbol read vector consist of b symbols, where b is an integer greater or equal to one. We start by defining basic properties.

Definition 3.3.1 (b-Symbol Read Vector)
Let $b \geq 1$ and $\mathbf{x} \in \Sigma^n$ a code word of length n over some alphabet Σ. Then the b-symbol read vector corresponding to $\mathbf{x}$ is defined as

$$\pi_b(\mathbf{x}) = \Big((x_0, .., x_{b-1}), (x_1, ..., x_b), ..., (x_{n-1}, x_0, ..., x_{b-2})\Big),$$

where the indices are modulo n.

Similarly, as in the case of $b = 2$, we define the b-symbol distance and b-symbol weight.

Definition 3.3.2 (b-Symbol Distance)
The b-symbol distance between $\mathbf{x}, \mathbf{y} \in \Sigma^n$ is defined as

$$d_b(\mathbf{x}, \mathbf{y}) = d_H(\pi_b(\mathbf{x}), \pi_b(\mathbf{y})).$$

Definition 3.3.3
The b-symbol weight of $\mathbf{x} \in \Sigma^n$ is defined as

$$w_b(\mathbf{x}) = w_H(\pi_b(\mathbf{x})).$$

By the same argument as in the case of $b = 2$, we have that $d_b(\mathbf{x}, \mathbf{y}) = w_b(\mathbf{x} - \mathbf{y})$, for all $\mathbf{x}, \mathbf{y} \in \Sigma^n$. In the following, we will generalize the results of Proposition 3.1.7 for all $b \geq 1$.

Proposition 3.3.4 ([22])
Let $\mathbf{x} \in \Sigma^n$ be such that $0 < w_H(\mathbf{x}) \leq n - (b-1)$. Then,

$$w_H(\mathbf{x}) + b - 1 \leq w_b(\mathbf{x}) \leq b \cdot w_H(\mathbf{x}).$$

Proof. Let $S = \{i \ : \ x_i \neq 0\}$, which gives us $|S| = w_H(\mathbf{x})$. Since every symbol $x_i \in \mathbf{x}$ appears in b entries of $\pi_b(\mathbf{x})$, the number of non-zero entries of $\pi_b(\mathbf{x})$ is at most $|S| \cdot b = w_H(\mathbf{x}) \cdot b$. This gives the upper bound $w_b(\mathbf{x}) \leq b \cdot w_H(\mathbf{x})$.
For the lower bound, we consider two cases.

Case 1: $\mathbf{x}$ has no sequence of b consecutive zeros
Then, for every index i, we have $\pi_b(\mathbf{x})_i \neq 0$, and therefore $w_b(\mathbf{x}) = n$. This gives us

$$w_H(\mathbf{x}) + b - 1 \leq n - (b-1) + (b-1) = n = w_b(\mathbf{x}).$$

Case 2: $\mathbf{x}$ has a sequence of at least b consecutive zeros
Note that by assumption we have $0 < w_H(\mathbf{x})$, thus we can find a index k such that $x_k = x_{k+1} = ... = x_{k+b-1} = 0$ and $x_{k+b} \neq 0$. Therefore, for all $k+1 \leq i \leq k+b-1$, we have $\pi_b(\mathbf{x})_i \neq 0$. In addition to that, we have $\pi_b(\mathbf{x})_i \neq 0$ for all i where $x_i \neq 0$. This implies that

$$w_b(\mathbf{x}) \geq w_H(\mathbf{x}) + b - 1$$

□

From now on, we consider Σ as $\mathbb{F}_q$ for some prime power q. If we consider the b-weight and $(b+1)$-weight of a nonzero element in $\mathbb{F}_q^n$, we have the following statement.

Proposition 3.3.5 ([8])
For any nonzero vector $\mathbf{x} = (x_0, x_1, ..., x_{n-1})$ *in* $\mathbb{F}_q^n$ *and* $w_b(\mathbf{x}) < n$*, we have*

$$w_{b+1}(\mathbf{x}) \geq w_b(\mathbf{x}) + 1$$

Proof. We have clearly $w_{b+1}(\mathbf{x}) \geq w_b(\mathbf{x})$, since if $(x_i, ..., x_{i+b-1}) \neq \mathbf{0}$ then we also have $(x_i, ..., x_{i+b-1}, x_{i+b}) \neq \mathbf{0}$ for all $0 \leq i \leq n-1$, where the indices are considered modulo n. If $(x_i, ..., x_{i+b-1}) = \mathbf{0}$, then $x_{i+b} = 0$ changes nothing from the b-weight to the $(b+1)$-weight of $\mathbf{x}$, whereas if $x_{i+b} \neq 0$ we have $w_{b+1}(\mathbf{x}) \geq w_b(\mathbf{x}) + 1$. □

Example 3.3.6 ([8])
Consider the vector $\mathbf{v} = (1, 1, 1, 0, 0, 0, 0) \in \mathbb{F}_2^7$ *with Hamming weight* 3*. Then*

$$\pi_2(\mathbf{v}) = (11, 11, 10, 00, 00, 00, 01),$$

where we consider the entries ij *as* (i, j) *for* $i, j \in \mathbb{F}_2$*. This gives us Hamming weight* 4*, thus* $w_2(\mathbf{v}) = 4$*. If we look at the cases* $b = 3$ *and* $b = 4$ *we have*

$$\pi_3(\mathbf{v}) = (111, 110, 100, 000, 000, 001, 011)$$

$$\pi_4(\mathbf{v}) = (1110, 1100, 1000, 0000, 0001, 0011, 0111)$$

which give us 3*-symbol weight* $w_3(\mathbf{v}) = 5$ *and* 4*-symbol weight* $w_4(\mathbf{v}) = 6$*.*

Analog to the $b = 2$ case, we can generalize the concept of the minimum pair distance for all $b \geq 1$ to the minimum b-distance.

Definition 3.3.7 (Minimum b-Distance)
Let $\mathcal{C}$ *be a code over* $\mathbb{F}_q$ *of length* n*. Then the minimum* b*-distance of* $\mathcal{C}$ *is defined as*

$$d_b(\mathcal{C}) = \min\{d_b(\mathbf{x}, \mathbf{y}) \ : \ \mathbf{x}, \mathbf{y} \in \mathcal{C} \ , \ \mathbf{x} \neq \mathbf{y}\}.$$

Definition 3.3.8 ($(n, M, d_b)_q$ b-Symbol Code)
Let $\mathcal{C}$ *be a code over* $\mathbb{F}_q$ *of length* n*, size* M *and minimum* b*-distance* d_b*. Then we call* $\mathcal{C}$ *a linear* $(n, M, d_b)_q$ b*-symbol code.*

Note that the case $b = 1$ corresponds to the conventional codes and the case $b = 2$ to the symbol-pair codes. The following theorem will give us an upper bound for the size M of a b-symbol code.

Theorem 3.3.9 (Singleton Bound [8])
Let $q \geq 2$ *and* $b \leq d_b \leq n$*. If* $\mathcal{C}$ *is an* $(n, M, d_b)_q$ b*-symbol code, then we have*

$$M \leq q^{n-d_b+b}.$$

Proof. Suppose that $\mathcal{C}$ is an $(n, M, d_b)_q$ b-symbol code with $q \geq 2$ and $b \leq d_b \leq n$. Delete the last $d_b - b$ coordinates from all the code words in $\mathcal{C}$. Note that any $d_b - b$ consecutive coordinates contribute at most $d_b - 1$ to the b-distance of two code words. Thus, the resulting code words of length $n - d_b + b$ are still distinct since $\mathcal{C}$ has minimum b-distance d_b. Note that the maximum number of distinct vectors in $\mathbb{F}_q^{n-d_b+b}$ is q^{n-d_b+b}, which gives us the upper bound. □

Analog to the Singleton bound for the conventional code, we can define maximum distance separable b-symbol codes.

Definition 3.3.10 (MDS $(n, d_b)_q$ b-Symbol code)
A $(n, M, d_b)_q$ b-symbol code $\mathcal{C}$ which attains the Singleton bound $M = q^{n-d_b+b}$ is called a maximum distance separable (MDS) $(n, d_b)_q$ b-Symbol code.

Theorem 3.3.11 ([8])
A linear MDS $(n, d_b)_q$ b-symbol code $\mathcal{C}$ with $d_b < n$ is also an MDS $(n, d_b+1)_q$ $(b+1)$-symbol code.

Proof. Using the fact that for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, we have $d_b(\mathbf{x}, \mathbf{y}) = w_b(\mathbf{x}-\mathbf{y})$ and by Proposition 3.3.5 we have $w_{b+1}(\mathbf{x}) \geq w_b(\mathbf{x}) + 1$, we obtain therefore $d_{b+1} \geq d_b + 1$. Thus

$$|\mathcal{C}| = q^{n-d_b+b} \geq q^{n-d_{b+1}+b+1}.$$

□

Similar to the conventional MDS codes, the construction of MDS b-symbol codes is of great interest. One can find more details and examples, especially on MDS symbol-pair codes, in [10], [5], [6], and [7]. Regarding b-symbol MDS codes, one can find lots of constructions of b-symbol MDS codes in [8]. From now on, we will focus more on cyclic codes, where we will also refer to the b-symbol Singleton bound from Theorem 3.3.9. To do so, we need to investigate the b-symbol weights of cyclic codes more.

Chapter 4

b-Symbol Weights of Cyclic Codes

In this Chapter, we will focus on finding ways to compute the b-symbol weight of code words of some cyclic code over some finite field $\mathbb{F}_q$. We will see that these cyclic codes are semiprimitive irreducible cyclic codes. We recall that for any element $\mathbf{a} = (a_0, a_1, ..., a_{n-1})$ in $\mathbb{F}_q^n$, we have the polynomial representation of $\mathbf{a}$ as $a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathbb{F}_q[x]$. Considering a cyclic code $\mathcal{C}$ of length n over $\mathbb{F}_q^n$, we can identify this code as an ideal I of $R = \mathbb{F}_q[x]/ < x^n - 1 >$. Since R is a principal ideal domain, there exists a polynomial $g(x) \in \mathbb{F}_q[x]$ of smallest degree such that $g(x) + < x^n - 1 > \in I$. We call this polynomial the generator polynomial of the cyclic code $\mathcal{C}$. This will be useful since we want to use some function field theory to obtain the b-weights of some cyclic codes. We start this Chapter by introducing the trace representation of a cyclic code. Later, we will use the same ideas to establish a trace representation of b-symbol codes.

4.1 Trace Representation of Cyclic Codes

In this section, we want to show that we can represent cyclic codes as trace codes. Since we can describe code words of cyclic codes as polynomials, we will use some function field theory from section 2.2.
First, we introduce some useful codes.

Definition 4.1.1 ($C(D, V)$ [18])
Let F be an algebraic function field over the constant field $\mathbb{F}_{q^m}$ and let $V \subseteq F$ be a finite-dimensional $\mathbb{F}_{q^m}$-subspace of F. Let $P_1, ..., P_n \in \mathbb{P}_F$ be n distinct places of degree one such that $v_{P_i}(f) \geq 0$ for all $f \in V$ and $i = 1, ..., n$. We consider the divisor $D = P_1 + \cdots + P_n$ and define

$$C(D, V) := \{(f(P_1), ..., f(P_n)) \ : \ f \in V\} \subseteq (\mathbb{F}_{q^m})^n.$$

We can define the trace code of $C(D, V)$ by considering the trace of each symbol in a code word. We obtain the following definition.

Definition 4.1.2 ($Tr_D(V)$ [18])
Let V and D be as before, then we define the trace code of $C(D, V)$ with respect to the

extension $\mathbb{F}_{q^m}/\mathbb{F}_q$ *as*

$$Tr_D(V) := \{(Tr(f(P_1)), ..., Tr(f(P_n))) \ : \ f \in V\} = Tr(C(D,V)) \subseteq \mathbb{F}_q^n.$$

A key observation of the definitions above is that $C(D,V)$ is a code over $(\mathbb{F}_{q^m})^n$, whereas $Tr_D(V)$ is a code over $(\mathbb{F}_q)^n$. Further, one can show that every code $\mathcal{C} \subseteq (\mathbb{F}_q)^n$ can be represented as $\mathcal{C} = Tr_D(V)$ for some V and D.

Proposition 4.1.3 ([18])
Let $F = \mathbb{F}_{q^m}(z)$ *be the rational function field over* $\mathbb{F}_{q^m}$*. Then for every code* $\mathcal{C} \subseteq (\mathbb{F}_q)^n$ *one can find suitable* $V \subseteq F$ *and divisor* D *such that*

$$\mathcal{C} = Tr_D(V).$$

Proof. Let $m \in \mathbb{N}$ be such that $q^m \geq n$. Let $F = \mathbb{F}_{q^m}(z)$. Let $\alpha_1, ..., \alpha_n$ be distinct elements in $\mathbb{F}_{q^m}$. We denote $P_i \in \mathbb{P}_F$ the rational place corresponding to the zero of $z - \alpha_i$ and let $D = P_1 + \cdots + P_n$. For a given code $\mathcal{C}$, we choose a basis $\{a^{(1)}, ..., a^{(k)}\}$, where $a^{(j)} = (a_1^{(j)}, ..., a_n^{(j)})$. For $j = 1, ..., k$ we choose polynomials $f_j \in \mathbb{F}_{q^m}[z]$ with the property $f_j(\alpha_i) = a_i^{(j)}$ for $i = 1, ..., n$. Let $V \subseteq F$ be the $\mathbb{F}_{q^m}$-linear subspace generated by $f_1, ..., f_k$. Then, by the construction of V and D, we have that $\mathcal{C} = Tr_D(V)$. This can be easily seen by the fact that $Tr(f_j(\alpha_i)) = Tr(a_i^{(j)}) = m \cdot a_i^{(j)}$. □

The goal is now to represent any cyclic code $\mathcal{C}$ as a trace code $Tr_D(V)$ for suitable V and D. Let $\mathcal{C}$ be a cyclic code of length n over $\mathbb{F}_q$ with the generator polynomial $g(x)$. Let $m \geq 1$ be the smallest integer satisfying $q^m \equiv 1 \mod n$ (i.e. m is the multiplicative order of q modulo n). Then the polynomial $x^n - 1$ factors over $\mathbb{F}_{q^m}$ as

$$x^n - 1 = \prod_{\nu=0}^{n-1} (x - \eta^\nu), \tag{4.1}$$

where $\eta \in \mathbb{F}_{q^m}$ is a primitive n-th root of 1. If we further assume that $\gcd(n, q) = 1$, then all linear factors in (4.1) are distinct. Using the fact that $g(x) \mid x^n - 1$, we can write the generator polynomial as

$$g(x) = \prod_{\nu \in I} (x - \eta^\nu), \tag{4.2}$$

where I is a subset of $\{0, ..., n-1\}$. Notice that I consists of all elements $\nu \in \{0, 1, ..., n-1\}$ such that

$$c(x) \in \mathcal{C} \iff c(\eta^\nu) = 0 \ \text{ for all } \nu \in I. \tag{4.3}$$

We can weaken the condition in (4.3) by considering cyclotomic cosets.

Definition 4.1.4 (Cyclotomic Coset)
We define for an integer $0 \leq i \leq n-1$

$$C(i) := \{j \in \mathbb{Z} \ : \ 0 \leq j \leq n-1 \ \text{ and } j \equiv q^l i \mod n \ \text{ for some } l \geq 0\}$$

as the cyclotomic coset of i*.*

It is easy to see that the cyclotomic cosets are pairwise disjoint and

$$\{0, 1, ..., n-1\} = \bigcup_{t=1}^{s} C(i_t),$$

where $s \leq n$. Further, if we take exactly one element of each cyclotomic coset, we can define a set of cyclotomic coset representatives.

Definition 4.1.5 (Set of Cyclotomic Coset Representatives)
Let $M \subseteq \{0, 1, ..., n-1\}$. We call $M_0 \subseteq \mathbb{Z}$ a complete set of cyclotomic coset representatives of M if for each $\nu \in M$ there exists a unique $\nu_0 \in M_0$ such that $\nu_0 \mod n \in C(\nu)$.

We can weaken the condition in (4.3) by replacing $I \subseteq \{0, 1, ..., n-1\}$ with its set of cyclotomic cosets representatives I_0.

$$c(x) \in \mathcal{C} \iff c(\eta^\nu) = 0 \text{ for all } \nu \in I_0. \tag{4.4}$$

Consider the dual code $\mathcal{C}^\perp$ of $\mathcal{C}$. We know that the check polynomial of $\mathcal{C}$ is given by

$$h(x) = (x^n - 1)/g(x) \in \mathbb{F}_q[x].$$

The reciprocal polynomial of $h(x)$ is given by

$$h^\perp(x) := h(0)^{-1} \cdot x^{\deg h(x)} \cdot h(x^{-1}).$$

Moreover, $h^\perp(x)$ is the generator polynomial of $\mathcal{C}^\perp$. We write

$$h^\perp(x) = \prod_{\rho \in J} (x - \eta^\rho) \text{ with } J \subseteq \{0, 1, ..., n-1\}.$$

We have that

$$\begin{aligned} \rho \in J &\iff h^\perp(\eta^\rho) = 0 \iff h(\eta^{-\rho}) = 0, \\ &\iff g(\eta^{-\rho}) \neq 0 \iff \rho \not\equiv -\nu \mod n \text{ for all } \nu \in I. \end{aligned}$$

Let J_0 be the set of cyclotomic coset representatives of J. Analog to (4.4) we have that

$$a(x) \in \mathcal{C}^\perp \iff a(\eta^\rho) = 0 \text{ for all } \rho \in J_0. \tag{4.5}$$

Note that the equation $a(\eta^\rho) = 0$ is equivalent to

$$\langle (a_0, a_1, ..., a_{n-1}), (1, \eta^\rho, \eta^{2\rho}, ..., \eta^{(n-1)\rho}) \rangle = 0, \tag{4.6}$$

where $(a_0, a_1, ..., a_{n-1})$ are the coefficients of $a(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$ and $\langle \cdot, \cdot \rangle$ is the canonical inner product on $(\mathbb{F}_{q^m})^n$.
Now, we are ready to present a trace representation of any cyclic code. Consider the rational function field $F = \mathbb{F}_{q^m}(z)$. Let $V \subseteq F$ be the $\mathbb{F}_{q^m}$-vector space generated by the set $\{z^\rho : \rho \in J_0\}$. Let $P_i \in \mathbb{P}_F$ be the rational places corresponding to the zeros of $z - \eta^{i-1}$ and set $D = P_1 + \cdots + P_n$. Then, by definition, $C(D, V)$ is a $\mathbb{F}_{q^m}$-vector space generated by the vectors $(1, \eta^\rho, \eta^{2\rho}, ..., \eta^{(n-1)\rho})$ with $\rho \in J_0$. Using (4.5) and (4.6), we conclude that

$$\mathcal{C}^\perp = C(D, V)^\perp|_{\mathbb{F}_q}.$$

In order to describe $\mathcal{C}$ as a trace code, we can use Delsarte's theorem.

Theorem 4.1.6 (Delsarte's Theorem [18])
For a code $\mathcal{C}$ over $\mathbb{F}_{q^m}$, we have that

$$(\mathcal{C}|_{\mathbb{F}_q})^\perp = Tr(\mathcal{C}^\perp).$$

Using Delsarte's theorem, we obtain that

$$\mathcal{C} = (C(D,V)^\perp|_{\mathbb{F}_q})^\perp = Tr(C(D,V)) = Tr_D(V).$$

We summarize all in the following proposition

Proposition 4.1.7 ([18])
Let $\mathcal{C}$ be a cyclic code of length n over $\mathbb{F}_q$ with generator polynomial $g(x)$. Let $\mathbb{F}_{q^m} = \mathbb{F}_q(\eta)$ where η is a n-th root of unity. Let $J = \{0 \leq \rho \leq n-1 \ : \ g(\eta^{-\rho}) \neq 0\}$ and J_0 the set of cyclotomic coset representatives of J.
Let $F = \mathbb{F}_{q^m}(z)$. We define $V \subseteq F$ as the $\mathbb{F}_{q^m}$-vector space generated by $\{z^\rho \ : \ \rho \in J_0\}$. Let $P_i \in \mathbb{P}_F$ be the rational place corresponding to the zero of $(z - \eta^{i-1})$ for $i = 1, ..., n$ and let $D = P_1 + \cdots + P_n$. Then we have

$$\mathcal{C} = Tr_D(V).$$

In the following section, we will see some concrete examples of Proposition 4.1.7 and eventually find a way to represent b-symbol codes as trace codes.

4.2 Trace Representation of $\pi_b(\mathcal{C})$

For the following, we assume that

- $\mathcal{C}$ is a cyclic code over $\mathbb{F}_q$ of length n and dimension $k = \dim_{\mathbb{F}_q}(\mathcal{C})$.
- n is an integer with $n \geq 3$ such that $gcd(n, q) = 1$.
- b is an integer with $2 \leq b \leq n-1$.
- r is an integer with $r \geq 2$ such that $n \mid (q^r - 1)$.
- $\eta \in \mathbb{F}_{q^r}^*$ is a primitive n-th root of 1.

Note that r exists because we assume $gcd(n, q) = 1$. Let $g \in \mathbb{F}_q[z]$ be the generator polynomial of $\mathcal{C}$. From section 2.3.3, we know that $\deg g(z) = n - k$ and $g(z) \mid (z^n - 1)$ in $\mathbb{F}_q[z]$. Since $n \mid (q^r - 1)$ and $g(z)$ has no repeated roots, we have that $g(z)$ splits into its linear factors over $\mathbb{F}_{q^r}$. Now, let us fix some sets concerning the zeros and nonzeros of $g(z)$. Let

$$\begin{aligned} S &= \{i \in \{0, 1, 2, ..., n-1\} \ : \ g(\eta^i) = 0\} \subseteq \{0, 1, ..., n-1\}, \\ \tilde{S} &= \{0, 1, 2, ..., n-1\} \setminus S, \\ U &= \{-j \mod n \ : \ j \in \tilde{S}\}. \end{aligned}$$

Definition 4.2.1 (Nonzero Set of $\mathcal{C}$ [17])
Let S, $\tilde{S}$ and U be as above.
Then, we call U the nonzero set of $\mathcal{C}$.

Notice that the nonzero set U is the same as the set J from Proposition 4.1.7. The following examples illustrate what the sets S, $\tilde{S}$ and U look like.

Example 4.2.2 ([17])
Let $q = 4$, $n = 21$ and $r = 3$. Let $\eta \in \mathbb{F}_{4^3}^*$ be a primitive 21-th root of unity. We let η be the root of $z^6 + z^5 + z^4 + z^2 + 1 \in \mathbb{F}_2[z]$, and we let

$$g(z) = (z - \eta^9)(z - \eta^{15})(z - \eta^{18})(z - \eta^5)(z - \eta^{20})(z - \eta^{17})$$
$$(z - \eta^{10})(z - \eta^{19})(z - \eta^{13})(z - \eta^7)$$

Let $\theta \in \mathbb{F}_4$ with $\theta^2 + \theta + 1 = 0$. Going through $g(z)$, we obtain that

$$g(z) = z^{10} + \theta z^9 + \theta z^8 + \theta^2 z^7 + z^6 + \theta z^5 + \theta^2 z^4 + z^2 + \theta z + \theta^2.$$

By choosing $\mathcal{C}$ such that g is its generator polynomial, We get the sets

$$S = \{5, 7, 9, 10, 13, 15, 17, 18, 19, 20\},$$

$$\tilde{S} = \{0, 1, 2, 3, 4, 6, 8, 11, 12, 14, 16\}.$$

Thus, we obtain the nonzero set of $\mathcal{C}$, namely

$$U = \{0, 5, 7, 9, 10, 13, 15, 17, 18, 19, 20\}.$$

Next, we want to find a trace representation of a cyclic code using Proposition 4.1.7. We recall some general facts about Galois groups.

Definition 4.2.3 (Frobenius Automorphism)
Let $r \geq 1$ and consider the fields $\mathbb{F}_q \subseteq \mathbb{F}_{q^r}$. The Frobenius automorphism is defined by

$$\varphi : \mathbb{F}_{q^r} \longrightarrow \mathbb{F}_{q^r} \; ; \quad \alpha \mapsto \alpha^q.$$

It is relatively easy to see that the Frobenius automorphism is, in fact, an automorphism. Since $\mathbb{F}_{q^r}$ is a field, it follows that the kernel of φ is 0, and since $\mathbb{F}_{q^r}$ is finite, φ must be surjective, which implies that φ is an isomorphism.

Proposition 4.2.4 ($Gal(\mathbb{F}_{q^r}/\mathbb{F}_q)$ **[18] [11])**
The field extension $\mathbb{F}_{q^r}/\mathbb{F}_q$ is a Galois extension of degree r. The Galois group $Gal(\mathbb{F}_{q^r}/\mathbb{F}_q)$ is cyclic and generated by the Frobenius automorphism φ.

Proof. Let G be the group generated by the Frobenius automorphism φ. Notice that for any $x, y \in \mathbb{F}_{q^r}$ we obtain, using the Freshman's dream, $\varphi(x + y) = (x + y)^q = x^q + y^q$. Additionally, we have $\varphi(x \cdot y) = (x \cdot y)^q = x^q \cdot y^q = \varphi(x) \cdot \varphi(y)$. Thus, φ is indeed a homomorphism from $\mathbb{F}_{q^r}$ to itself. Further, notice that

$$\varphi^r(x) = x^{q^r} = x$$

for all x in $\mathbb{F}_{q^r}$. I.e. $\varphi^r = id$. Let $d \geq 1$ be the smallest integer such that $\varphi^d = id$. Then $\varphi^d(x) = x^{q^d} = x$ and each $x \in \mathbb{F}_{q^r}$ is a root of the equation

$$X^{q^d} - X = 0.$$

This equation has at most q^d roots, which implies that $d \geq r$, and therefore we conclude $d = r$. Since the size of $Gal(\mathbb{F}_{q^r}/\mathbb{F}_q)$ is $[\mathbb{F}_{q^r} : \mathbb{F}_q] = r$, we have $Gal(\mathbb{F}_{q^r}/\mathbb{F}_q) =< \varphi >= G$.
□

We consider now the following action of the cyclic group $Gal(\mathbb{F}_{q^m}/\mathbb{F}_q)$ on the set $\{0, 1, ..., n-1\}$. For $u \in \{0, 1, ..., n-1\}$ the action of the Frobenius automorphism is given by $uq \mod n \in \{0, 1, ..., n-1\}$.

Definition 4.2.5 (q-Cyclotomic Coset Modulo n)
Let $u \in \{0, 1, ..., n-1\}$. We call the orbit of u,

$$\{u^i \mod n \ : \ 0 \leq i \leq r-1\},$$

the q-cyclotomic coset of u modulo n.

We call any set $A \subseteq \{0, 1, ..., n-1\}$ closed if $u \in A$ implies that $uq \mod n \in A$. Further, any closed $A \subseteq \{0, 1, ..., n-1\}$ is a disjoint union of q-cyclotomic cosets modulo n. Using the fact that the nonzero set U is closed, we obtain that U is the disjoint union of q-cyclotomic cosets modulo n. By choosing one element of each q-cyclotomic coset, we can define the basic nonzero set of a cyclic code $\mathcal{C}$.

Definition 4.2.6 (Basic Nonzero Set of $\mathcal{C}$)
Let U be the nonzero set of a cyclic code $\mathcal{C}$. Let U_0 be a subset of U where U_0 contains exactly one element from every q-cyclotomic coset of U. Then we call U_0 a basic nonzero set of $\mathcal{C}$.

Notice that the basic nonzero set U_0 is the same as the J_0 from Proposition 4.1.7. From the definition, we see that the basic nonzero set U_0 is not uniquely determined. We will see this fact in the following example.

Example 4.2.7 ([17])
Analog to the example before, we choose $q = 4$, $n = 21$, and $r = 3$. Let $\mathcal{C}$ be the same code as in example 4.2.2. Then the 4-cyclotomic cosets modulo 21 are

$$\bar{0} = \{0\},\ \bar{1} = \{1, 4, 16\},\ \bar{2} = \{2, 8, 11\},\ \bar{3} = \{3, 12, 6\},\ \bar{5} = \{5, 20, 17\}$$
$$\bar{7} = \{7\},\ \bar{9} = \{9, 15, 18\},\ \bar{10} = \{10, 19, 13\},\ \bar{14} = \{14\},$$

and the nonzero set of $\mathcal{C}$ is

$$U = \{0, 5, 7, 9, 10, 13, 15, 17, 18, 19, 20\} = \bar{0} \sqcup \bar{5} \sqcup \bar{7} \sqcup \bar{9} \sqcup \bar{10}.$$

Therefore, we can define a basic nonzero set of $\mathcal{C}$ as

$$U_0 = \{0, 5, 7, 9, 10\}.$$

Next, we want to define the trace representation of a cyclic code. Recall that the trace map is defined by

$$Tr : \mathbb{F}_{q^r} \to \mathbb{F}_q,\ x \mapsto x + x^q + \cdots x^{q^{r-1}}.$$

Further, the trace map is surjective and a $\mathbb{F}_q$-linear map.

Definition 4.2.8 (Trace Representation of a Cyclic Code)
Let $U_0 = \{u_1, u_2, ..., u_l\}$ be a basic nonzero set of a cyclic code $\mathcal{C}$, and let $P(U_0)$ denote the $\mathbb{F}_{q^r}$-linear subspace of $\mathbb{F}_{q^r}[x]$, which we define through

$$P(U_0) = \{a_1 x^{u_1} + \cdots + a_l x^{u_l} \ : \ a_1, ..., a_l \in \mathbb{F}_{q^r}\}.$$

For any $f(x) \in P(U_0)$, let

$$\underline{Tr_n(f(x))} := \big(Tr(f(\eta^0)), Tr(f(\eta^1)), ..., Tr(f(\eta^{n-1}))\big).$$

Then the trace representation of $\mathcal{C}$ is given by

$$\mathcal{C} = \{\underline{Tr_n(f(x))} \ : \ f \in P(U_0)\}$$

The underline with the integer n in $\underline{Tr_n(f(x))}$ emphasizes that we are considering the n traces $Tr(f(\eta^0)), ..., Tr(f(\eta^{(n-1)}))$. Notice that $P(U_0)$ equals the $\mathbb{F}_{q^r}$-vector space V from Proposition 4.1.7. For any code word $c \in \mathcal{C}$, one can find $f \in P(U_0)$ such that $c = \underline{Tr_n(f(x))} \in \mathbb{F}_q^n$. Therefore, we also denote $\underline{Tr_n(f(x))}$ as

$$c(f) = \big(Tr(f(\eta^0)), Tr(f(\eta^1)), ..., Tr(f(\eta^{n-1}))\big).$$

We can now advance example 4.2.7 and find a trace representation of the corresponding cyclic code.

Example 4.2.9 ([17])
From example 4.2.2 and example 4.2.7, we have $U_0 = \{0, 5, 7, 9, 10\}$ as a basic set of $\mathcal{C}$. This gives us the set

$$P(U_0) = \{a_0 + a_5x^5 + a_7x^7 + a^9x^9 + a_{10}x^{10} \ : \ a_0, a_5, a_7, a_9, a_{10} \in \mathbb{F}_{4^3}.\}$$

For any code word $\mathbf{c} = (c_0, c_1, ..., c_{20}) \in \mathcal{C} \subseteq \mathbb{F}_4^{21}$ there exists $f \in P(U_0)$ such that $\mathbf{c} = c(f)$. In other words, there exists $a_0, a_5, a_7, a_9, a_{10} \in \mathbb{F}_{4^3}$ such that for all $0 \leq i \leq 20$

$$c_i = Tr(a_0 + a_5\eta^{5i} + a_7\eta^{7i} + a_9\eta^{9i} + a_{10}\eta^{10i}).$$

Since we have defined a trace representation of a cyclic code, the next step is to find a trace representation of the corresponding b-symbol code.

Definition 4.2.10
Let $0 \leq t \leq n-1$ and $f \in P(U_0)$. We define $f^{(t)}(x)$ to be the polynomial

$$f^{(t)}(x) = f(\eta^t x) \in P(U_0).$$

Using this definition, we define a trace representation of the corresponding b-symbol code of a cyclic code of length n in the following theorem.

Theorem 4.2.11 (Trace Representation of $\pi_b(\mathcal{C})$, [17])
Let $\mathcal{C}$ be a cyclic code over $\mathbb{F}_q$ of length n such that $gcd(n, q) = 1$. Let $2 \leq b \leq n-1$ be an integer. For the corresponding b-symbol code $\pi_b(\mathcal{C})$ of length n over $\mathbb{F}_q^b$ we have

$$\pi_b(\mathcal{C}) = \Big\{\Big(\underline{Tr_b(f(\eta))}, \underline{Tr_b(f^{(1)}(\eta))}, \cdots, \underline{Tr_b(f^{(n-1)}(\eta))}\Big) \ : \ f \in P(U_0)\Big\}.$$

For any element $(\beta_0, ..., \beta_{n-1}) \in \pi_b(\mathcal{C})$, there exists $f \in P(U_0)$, such that for all $0 \leq i \leq n-1$ we have

$$\beta_i = \Big(Tr(f(\eta^i)), Tr(f^{(1)}(\eta^i)), ..., Tr(f^{(b-1)}(\eta^i))\Big) \in \mathbb{F}_n^b.$$

Note that we use b instead of n in $\underline{Tr_b(f^{(i)}(\eta))}$ since we consider b consecutive symbols.

Proof. Let $f \in P(U_0)$ and $c(f) = (c_0, c_1, ..., c_{n-1})$ be the corresponding code word. Let $\pi_b(c(f))$ be the corresponding b-symbol code word in $\pi_b(\mathcal{C})$. For $\pi_b(c(f)) = (\beta_0, \beta_1, ..., \beta_{n-1})$, we have $\beta_i = (c_i, c_{i+1}, ..., c_{i+b-1})$ where the indices are modulo n. Notice that for $0 \leq l \leq b-1$, we have $c_{i+l} = Tr(f(\eta^{i+l})) = Tr(f(\eta^l \eta^i)) = Tr(f^{(l)}(\eta^i))$. This gives us

$$\beta_i = \Big(Tr(f(\eta^i)), Tr(f^{(1)}(\eta^i)), ..., Tr(f^{(b-1)}(\eta^i))\Big) = \underline{Tr_b(f^{(i)}(\eta))},$$

which completes the proof. □

For instance, we can now consider the trace representation of the b-symbol code of the code in example 4.2.2.

Example 4.2.12 ([17])
Let again $q = 4$, $n = 21$ and $r = 3$. Let $\mathcal{C}$ be the cyclic code of length 21 over $\mathbb{F}_4$ as we have seen in example 4.2.2, 4.2.7 and 4.2.9. Let us consider $b = 3$. We want to find the trace representation of the 3-symbol code of $\mathcal{C}$. We know that $f \in P(U_0)$ has the form

$$f(x) = a_0 + a_5 x^5 + a_7 x^7 + a_9 x^9 + a_{10} x^{10} \in P(U_0).$$

Hence for any element $(\beta_0, \beta_1, ..., \beta_{20})$ in $\mu_3(\mathcal{C})$, there exists $a_0, a_5, a_7, a_9, a_{10} \in \mathbb{F}_{4^3}$ such that for $0 \leq i \leq 20$ we have

$$\begin{aligned}\beta_i = \Big(&Tr(a_0 + a_5\eta^{5i} + a_7\eta^{7i} + a_9\eta^{9i} + a_{10}\eta^{10i}),\\ &Tr(a_0 + a_5\eta^{5i+5} + a_7\eta^{7i+7} + a_9\eta^{9i+9} + a_{10}\eta^{10i+10}),\\ &Tr(a_0 + a_5\eta^{5i+10} + a_7\eta^{7i+14} + a_9\eta^{9i+18} + a_{10}\eta^{10i+20})\Big).\end{aligned}$$

Moreover, we see that

$$\beta_i = \Big(Tr(f(\eta^i)), Tr(f^{(1)}(\eta^i)), Tr(f^{(2)}(\eta^i)\Big) = \underline{Tr_3(f^{(i)}(\eta))}.$$

We've found a way to write b-symbol codes as trace codes. The next big step is to find the b-symbol weight distribution of some cyclic codes. We are now able to use some properties of the trace map that are helpful in this matter. The most important one is that for all $y \in \mathbb{F}_{q^r}$

$$Tr(y) = 0 \Longleftrightarrow x^q - x = y \text{ for some } x \in \mathbb{F}_{q^r}.$$

Therefore, we will deduce some facts about the b-symbol weight by investigating the solutions of the curve $x - x^q = y$ over $\mathbb{F}_{q^r}$.

4.3 b-Symbol Weights of Cyclic Codes

Let again $\mathcal{C}$ be a cyclic code of length n such that n divides $q^r - 1$. For the next part, we keep the assumptions from the last section and add the following.

- The nonzero set of $\mathcal{C}$ is exactly one q-cyclotomic coset $U \neq \{0\}$ in $\mathbb{Z}/n\mathbb{Z}$.

Note that if $U = \{0\}$, then $\mathcal{C}$ is a repetition code which implies that any nonzero b-symbol code word has Hamming weight n for any $1 \leq b \leq n-1$.
Next, we will establish some useful facts about the b-symbol Hamming weight enumerator. Recall that the Hamming weight enumerator of a given code $\mathcal{C}$ is given by the polynomial

$$A(z) = A_0 + A_1 Z + \cdots + A_n Z^n, \quad \text{where } A_i = |\{\alpha \in \mathcal{C} \ : \ w_H(\alpha) = i\}|.$$

Similarly, we can define the b-symbol Hamming weight enumerator.

Definition 4.3.1 (b-Symbol Hamming Weight Enumerator)
The b-symbol Hamming weight enumerator of a given code $\mathcal{C}$ is given by the polynomial

$$A^{(b)}(z) = A_0^{(b)} + A_1^{(b)} z + \cdots + A_n^{(b)} z^n,$$

where $A_i^{(b)} = |\{\beta \in \pi_b(\mathcal{C}) \ : \ w_b(\beta) = i\}|$.

We call the vector $(A_0, A_1, ..., A_n)$ (resp. $(A_0^{(b)}, A_1^{(b)}, ..., A_n^{(b)})$) the Hamming weight distribution of $\mathcal{C}$ (resp. the b-symbol Hamming weight distribution of $\mathcal{C}$). In general, determining the Hamming weight distribution of an irreducible cyclic code is still a difficult open problem. One can find more on this in [7], but we will see that there is a connection between our codes in this section and irreducible cyclic codes.
We start by introducing some notations.

Definition 4.3.2
Let $u \in U \setminus \{0\}$. For $1 \leq t \leq n-1$ we set $V(t) = Span_{\mathbb{F}_q}\{1, \eta^u, ..., \eta^{(t-1)u}\}$. We define μ to be the largest positive integer t such that $\dim_{\mathbb{F}_q} V(t) = t$.

We show now that μ is independent of the choice of the primitive n-th root of unity η and the choice of the element u of the nonzero set U.

Lemma 4.3.3 ([17])
Let $\mathbb{F}_q(\eta^u)$ denote the smallest finite field extension of $\mathbb{F}_q$ containing η^u. Using the same notations as in definition 4.3.2, we have

$$\mu = \dim_{\mathbb{F}_q} \mathbb{F}_q(\eta^u).$$

Proof. Note that for consecutive dimensions, we have

$$\dim_{\mathbb{F}_q} V(t+1) - \dim_{\mathbb{F}_q} V(t) \in \{0, 1\},$$

for all t. Since μ is by definition the largest positive integer such that $dim_{\mathbb{F}_q} V(t) = t$, we have that $dim_{\mathbb{F}_q} V(\mu+1) - \dim_{\mathbb{F}_q} V(\mu) = 0$. This implies that μ is the smallest positive integer t such that

$$\eta^{u(t+i)} \in Span_{\mathbb{F}_q}\{1, \eta^u, ..., \eta^{u(t-1)}\}$$

for all $i \geq 0$. In other words, μ is the smallest positive integer t such that

$$\mathbb{F}_q[\eta^u] \subseteq Span_{\mathbb{F}_q}\{1, \eta^u, ..., \eta^{u(t-1)}\}.$$

This implies that $\mathbb{F}_q(\eta^u) = \mathbb{F}_q[\eta^u] = Span_{\mathbb{F}_q}\{1, \eta^u, ..., \eta^{u(t-1)}\}$. □

As a direct consequence of Lemma 4.3.3, we have multiple characterizations of μ.

Corollary 4.3.4 ([17])
Using the same notations as above, the following characterizations of μ are equivalent.

- $\mu = \dim_{\mathbb{F}_q} \mathbb{F}_{||}(\eta^u)$.
- $\mu = \dim_{\mathbb{F}_q} \mathcal{C}$.
- *μ is the multiplicative order of $q \mod \frac{n}{gcd(n,u)}$. I.e. μ is the smallest positive integer such that $q^\mu \equiv 1 \mod \frac{n}{gcd(n,u)}$.*
- *μ is the size of the q-cyclotomic coset U, containing u, in $\mathbb{Z}/n\mathbb{Z}$.*

Proof. Using the fact that η is a n-th root of unity, we have that the multiplicative order of η^u is $\frac{n}{gcd(n,u)}$. Hence we have $\dim_{\mathbb{F}_q} \mathbb{F}_q(\eta^u) = \mu$ if and only if μ is the smallest positive integer r such that $\frac{n}{gcd(n,u)}$ divides $q^r - 1$. This is equivalent to $q^r \equiv 1 \mod \frac{n}{gcd(n,u)}$. This proves the equivalence between the first and the third characterization.
Let U be the nonzero set of $\mathcal{C}$. By the definition of U, we have that for every $u \in U$, $g(\eta^u) \neq 0$, where $g(z)$ denotes the generator polynomial of $\mathcal{C}$. This means that $\dim_{\mathbb{F}_q} \mathcal{C}$ is the size of U. Note that U is the q-cyclotomic coset containing u in $\mathbb{Z}/n\mathbb{Z}$, and consists by assumption of exactly one q-cyclotomic coset. The size of U is the smallest positive integer r such that $q^r u \equiv u \mod n$. This implies that the size of U is the smallest positive integer r such that $\frac{n}{gcd(n,u)}$ divides $q^r - 1$. This implies the equivalence of the third and the fourth characterization of μ, and therefore, we also have that $\mu = \dim_{\mathbb{F}_q} \mathcal{C}$. □

Now, we are ready to state the stability theorem for the b-symbol Hamming weight enumerator of $\mathcal{C}$, where $\mathcal{C}$ satisfies all assumptions we made in this section. Later, we will generalize this theorem for arbitrary cyclic codes.

Theorem 4.3.5 (Weak Stability Theorem [17])
Assume that $gcd(n, q) = 1$. Let $\mathcal{C}$ be a cyclic code of length n such that its nonzero set U is exactly one q-cyclotomic coset of $\mathbb{Z}/n\mathbb{Z}$. Assume that $U \neq \{0\}$ and let $u \in U$. Let $k = \dim_{\mathbb{F}_q} \mathcal{C}$. For any integer b with $k \leq b \leq n - 1$, we have that the b-symbol Hamming weight enumerator of $\mathcal{C}$ is the same as the k-symbol Hamming weight enumerator of $\mathcal{C}$.

Proof. Let $f(x) = ax^u \in \mathbb{F}_{q^r}[x] \setminus \{0\}$ be an arbitrary nonzero polynomial in $P(\{u\}) = \{ax^u : a \in \mathbb{F}_{q^r}\}$. Let $\pi_k(c(f)) \in \pi_k(\mathcal{C})$ and $\pi_b(c(f)) \in \pi_b(c(f)) \in \pi_b(\mathcal{C})$ be some k-symbol resp. b-symbol code word over $\mathcal{C}$. We have

$$\pi_b(c(f)) = \big(\underline{Tr_b(f(\eta))}, \underline{Tr_b(f^{(1)}(\eta))}, ..., \underline{Tr_b(f^{(b)}(\eta))}\big).$$

We set $\pi_b(c(f)) = \big(c_0^{(b)}(f), c_1^{(b)}(f), ..., c_{n-1}^{(b)}(f)\big) \in (\mathbb{F}_q^b)^n$. For the symbols in $\pi_b(c(f))$, we have for $0 \leq i \leq n - 1$

$$c_i^{(b)}(f) = \big(Tr_b(a\eta^{ui}), Tr_b(\eta^u a\eta^{ui}), ..., Tr_b(\eta^{(b-1)u} a\eta^{ui})\big) \in \mathbb{F}_q^b.$$

Analog, we obtain the symbols of $\pi_k(c(f))$, namely

$$c_i^{(k)}(f) = \big(Tr_k(a\eta^{ui}), Tr_k(\eta^u a\eta^{ui}), ..., Tr_k(\eta^{(k-1)u} a\eta^{ui})\big) \in \mathbb{F}_q^k.$$

Since $k = \dim_{\mathbb{F}_q}(\mathcal{C})$, by Corollary 4.3.4 we obtain that

$$Q := Span_{\mathbb{F}_q}\{1, \eta^u, ..., \eta^{(k-1)u}\} = Span_{\mathbb{F}_q}\{1, \eta^u, ..., \eta^{(b-1)u}\}.$$

Therefore if $\alpha \in \mathbb{F}_{q^r}$, then

$$\begin{aligned} 0 = Tr(\alpha) = Tr(\eta^u \alpha) = \cdots = Tr(\eta^{(k-1)u}\alpha), \\ \Longleftrightarrow 0 = Tr(\alpha) = Tr(\eta^u \alpha) = \cdots = Tr(\eta^{(b-1)u}\alpha). \end{aligned}$$

This implies that $c_i^{(k)}(f)$ contributes to the Hamming weight of the code word $\pi_k(c(f))$ of length n over the alphabet $\mathbb{F}_q^k$ if and only if $c_i^{(b)}(f)$ contributes to the Hamming weight of the code word $\pi_b(c(f))$ of length n over the alphabet $\mathbb{F}_q^b$. Thus, the values of the Hamming weights of $\pi_k(c(f))$ and $\pi_b(c(f))$ are the same.

□

Notice that Theorem 4.3.5 has some useful applications. For instance, if we increase the integer b for b-symbol codes, if $b \geq k = \dim_{\mathbb{F}_q}(\mathcal{C})$, we don't get any advantage for error correction.
Notice also that we have proved Theorem 4.3.5 under many assumptions for some specific cyclic codes. These codes have some connections to irreducible cyclic codes.

Recall that a cyclic code is said to be irreducible if its parity check polynomial is irreducible over $\mathbb{F}_q$. There is also an equivalent definition of irreducible cyclic codes using the trace map.

Theorem 4.3.6 (Trace Representation of an Irreducible Cyclic Code [19])
Let r be the multiplicative order of $q \mod n$. Let η be a primitive n-th root of unity in $\mathbb{F}_{q^r}$. Then the set

$$V := \{\big(Tr(a\eta^0), Tr(a\eta^1), ..., Tr(a\eta^{n-1})\big) \ : \ a \in \mathbb{F}_{q^r}\}$$

is an (n, r) irreducible cyclic code over $\mathbb{F}_q$.

Proof. Observe that $\big(Tr(a\eta^{-1}\eta^0), Tr(a\eta^{-1}\eta^1), ..., Tr(a\eta^{-1}\eta^{n-1})\big)$ is a cyclic shift of $\big(Tr(a\eta^0), Tr(a\eta^1), ..., Tr(a\eta^{n-1})\big)$. Therefore, V is a cyclic code. Further, we have that r is the smallest positive integer, such that $n \mid q^r - 1$, which implies that η is in no subfield of $\mathbb{F}_{q^r}$. Therefore, η is a zero of an irreducible polynomial $h(x) = h_0 + h_1 x + \cdots + h_r x^r$ of degree r in $\mathbb{F}_q[x]$.
Let $\big(Tr(a\eta^0), Tr(a\eta^1), ..., Tr(a\eta^{n-1})\big) = (c_0, c_1, ..., c_{n-1})$ then we have

$$\sum_{i=0}^{r} c_i h_i = Tr(ah(\eta)) = Tr(0) = 0.$$

In other words, we have a parity check equation for the code V. □

Using the trace representation of irreducible cyclic codes, we can observe a connection between the codes we used in this section and irreducible cyclic codes. To see that, let $m = gcd(u, n)$ and let $\tilde{n} = n/m$ such that $\eta^{u\tilde{n}} = 1$. Therefore, $\eta^{ui} = \eta^{u(i+\tilde{n})}$ for all $i \geq 0$. For a code word $c = (c_0, c_1, \cdots, c_{n-1}) \in \mathcal{C}$ and a code word $\mu_b(c) = (c_0^{(b)}, c_1^{(b)}, \cdots, c_{n-1}^{(b)}) \in \pi_b(\mathcal{C})$, we define the shortenings to the $\tilde{n}$ first symbols as

$$\begin{aligned} \bar{c} &= (c_0, c_1, \cdots, c_{\tilde{n}-1}), \\ \bar{c}^{(b)} &= (c_0^{(b)}, c_1^{(b)}, \cdots, c_{\tilde{n}-1}^{(b)}). \end{aligned} \tag{4.7}$$

Let $\bar{C} = \{\bar{c} \ : \ c \in \mathcal{C}\} \subseteq \mathbb{F}_q^{\tilde{n}}$ and $\bar{C}^{(b)} = \{\bar{c} \ : \ \mu_b(c) \in \pi_b(\mathcal{C})\} \subseteq (\mathbb{F}_q^b)^{\tilde{n}}$. Moreover, using the fact that $\eta^{ui} = \eta^{u(i+\tilde{n})}$ for $i \geq 0$, we see that $\bar{C}^{(b)} = \mu_b(\bar{C})$. Further, we have the following relations between the Hamming weights of $c, \bar{c}, \mu_b(c), \mu_b(\bar{c})$.

$$w_H(\bar{c}) = \frac{1}{m} w_H(c) \ \text{ and } \ w_H(\bar{c}^{(b)}) = \frac{1}{m} w_H(c^{(b)}). \tag{4.8}$$

Let $\theta = \eta^u \in \mathbb{F}_{q^r}^*$. Then θ is a primitive $\tilde{n}$-th root of unity. Moreover, since we assumed in this section that $q^r \equiv 1 \mod n$, we have that r is the multiplicative order of q modulo $\tilde{n}$. Thus, we can apply Theorem 4.3.6 and observe that

$$\bar{C} = \{\big(Tr(a\theta^0), Tr(a\theta^1), ..., Tr(a\theta^{\tilde{n}-1})\big) \ : \ a \in \mathbb{F}_{q^r}\}$$

is the irreducible cyclic code of length $\tilde{n}$ over $\mathbb{F}_q$. Notice that the cyclic code $\mathcal{C}$ can be obtained from $\bar{C}$ by repeating m times the code words $\bar{c} \in \bar{C}$. In other words, we have

$$\mathcal{C} = \left\{(\bar{c}, \bar{c}, ..., \bar{c}) \ : \ \bar{c} \in \bar{C}\right\}.$$

This shows us there is a connection between our codes in this section and irreducible cyclic codes. The next step is to generalize the results from Theorem 4.3.5 for a larger class of irreducible cyclic codes. Later, we will try to find ways to determine the b-symbol weight distribution of some irreducible cyclic codes. By making some assumptions on the cyclic code, we will be able to completely determine its b-symbol weight distribution up to an invariant that we will call $\mu(b)$ resp. $\mu_{(i)}(b)$.

4.3.1 Generalization of Weak Stability Theorem (Theorem 4.3.5)

First, we want to remove all assumptions we made on the cyclic code $\mathcal{C}$. For this section, we only assume that $\mathcal{C}$ is a cyclic code of length n such that n divides $q^r - 1$. Let U be the nonzero set of $\mathcal{C}$ and $U_1, U_2, ..., U_\rho$ the distinct q-cyclotomic cosets of $\mathbb{Z}/n\mathbb{Z}$ included in U. This means we can write U as

$$U = U_1 \sqcup U_2 \sqcup \cdots \sqcup U_\rho,$$

where $\rho \geq 1$ and $\sqcup$ indicates that $U_1, U_2, ..., U_\rho$ are pairwise disjoint. Further, we assume that $U \neq \{0\}$, to avoid the trivial case, where $\mathcal{C}$ is a repetition code.
Let $u_j \in U_j$ and $k_j = |U_j|$ for $1 \leq j \leq \rho$. The $\mathbb{F}_q$-dimension k of the code $\mathcal{C}$ satisfies $k = k_1 + \cdots k_\rho$. Let $\eta \in \mathbb{F}_{q^r}^*$ be a primitive n-th root of unity. We define for $0 \leq t \leq n-1$ the vector $\mathbf{v_t} \in \mathbb{F}_{q^r}^\rho$ as

$$\mathbf{v_t} = [\eta^{tu_1}, \eta^{tu_2}, ..., \eta^{tu_\rho}]. \tag{4.9}$$

Further, for $1 \leq t \leq n-1$, we define the $\mathbb{F}_q$-linear subspace $V(t) \subseteq \mathbb{F}_{q^r}^\rho$ as

$$V(t) = Span_{\mathbb{F}_q}\{\mathbf{v_0}, \mathbf{v_1}, ..., \mathbf{v_{t-1}}\}.$$

Using these definitions, we want to present a useful lemma.

Lemma 4.3.7 ([17])
Under the above notation, we have

$$\dim_{\mathbb{F}_q} V(t) = \begin{cases} t & \text{if } \ 1 \leq t \leq k-1, \\ k & \text{if } \ k \leq t \leq n-1. \end{cases}$$

Proof. Recall that $\mathbb{F}_q(\eta^{u_j})$ denotes the smallest finite field extension over $\mathbb{F}_q$ containing η^{u_j}. Considering the index of this field extension, we have that $[\mathbb{F}_q(\eta^{u_j}) : \mathbb{F}_q] = k_j$. Let $m_j(x) \in \mathbb{F}_q[x]$ be the minimal polynomial of η^{u_j} over $\mathbb{F}_q$. Clearly, we have $\deg m_j(x) = k_j$ and the set $\{m_1(x), m_2(x), ..., m_\rho(x)\}$ consists of irreducible polynomials over $\mathbb{F}_q$ which are pairwise distinct.
We start by showing that $\dim V(k) = k$. Assume the contrary, and let $e_0, e_1, ..., e_{k-1} \in \mathbb{F}_q$ with some nonzero e_i such that

$$e_0\mathbf{v_0} + e_1\mathbf{v_1} + \cdots + e_{k-1}\mathbf{v_{k-1}} = \mathbf{0}. \tag{4.10}$$

Let $1 \leq j \leq \rho$ and consider the j-th coordinates of equation (4.10),

$$e_0 + e_1\eta^{u_j} + e_2\eta^{2u_j} + \cdots + e_{k-1}\eta^{(k-1)u_j} = 0. \tag{4.11}$$

Let $h(x) = e_0 + e_1x + e_2x^2 + \cdots + e_{k-1}x^{k-1} \in \mathbb{F}_q[x]$, which is a nonzero polynomial of degree at most $k-1$. Equation (4.11) implies that η^{u_j} is a root of $h(x)$. We conclude that $h(\eta^{u_j}) = 0$ for all $1 \leq j \leq \rho$. Note that $m_j(x)$ is the minimal polynomial of η^{u_j} over $\mathbb{F}_q$ and therefore we have that $m_j(x)$ divides $h(x)$ for each $1 \leq j \leq \rho$. Using the fact that the set $\{m_1(x), m_2(x), ..., m_\rho(x)\}$ consists of irreducible polynomials over $\mathbb{F}_q$, which are also pairwise distinct, we obtain the fact that $\prod_{j=1}^{\rho} m_j(x) \mid h(x)$. This implies that

$$\deg h(x) \geq \sum_{j=1}^{\rho} m_j(x) = \sum_{j=1}^{\rho} k_j = k.$$

This contradicts the fact that $h(x)$ is a nonzero polynomial of degree at most $k-1$.
It is trivial to see that $V(t-1) \subseteq V(t)$ and

$$0 \leq \dim_{\mathbb{F}_q} V(t) - \dim_{\mathbb{F}_q} V(t-1) \leq 1 \tag{4.12}$$

for each $2 \leq t \leq n-1$. Notice that for $t = 1$, we have $V(1) = Span_{\mathbb{F}_q}\{[1, ..., 1]\}$ and therefore $\dim_{\mathbb{F}_q} V(1) = 1$. Combining the inequality (4.12) and the facts $\dim_{\mathbb{F}_q} V(1) = 1$, $\dim_{\mathbb{F}_q} V(k) = k$, we must have that $\dim_{\mathbb{F}_q} V(t) = t$ for all integers $1 \leq t \leq k$. This also shows that $\{\mathbf{v_0}, \mathbf{v_1}, ..., \mathbf{v_t}\}$ is a basis of $V(t)$ for each integer $1 \leq t \leq k$.
For the second part of the proof, we need to show that $V(k+i) \subseteq V(k)$ for $1 \leq i \leq n-k-1$. We show that by induction on i. Let $i = 1$ and $m(x) = m_1(x)m_2(x)\cdots m_\rho(x)$, which is a monic polynomial of degree k. We write $m(x)$ as

$$m(x) = x^k + e_{k-1}x^{k-1} + \cdots + e_1x + e_0,$$

where $e_0, e_1, ..., e_{k-1} \in \mathbb{F}_q$ are the coefficients of $m(x)$. Since $m(\eta^{u_j}) = 0$ for each $1 \leq j \leq \rho$, we must have that

$$\mathbf{v_k} + e_{k-1}\mathbf{v_{k-1}} + \cdots + e_1\mathbf{v_1} + e_0\mathbf{v_0} = \mathbf{0}.$$

This implies that $\mathbf{v_k} \in V(k)$ and therefore $V(k+1) \subseteq V(k)$. Now, we assume the induction hypothesis that $V(k+i) \subseteq V(k)$. Let $h(x) = x^i m(x)$, which is a monic polynomial of degree $k+i$. Let

$$h(x) = x^{k+i} + e_{k+i-1}x^{k+i-1} + \cdots + e_1x + e_0,$$

where $e_{k+i-1}, ..., e_1, e_0 \in \mathbb{F}_q$ are the coefficients of $h(x)$. Again, using the fact that $m(\eta^{u_j}) = 0$ for all $1 \leq j \leq \rho$, we have that

$$\mathbf{v_{k+i}} + e_{k+i-1}\mathbf{v_{k+i-1}} + \cdots + e_1\mathbf{v_1} + e_0\mathbf{v_0} = \mathbf{0}.$$

This implies that $\mathbf{v_{k+i}} \in V(k+i)$ and hence $V(k+i+1) \subseteq V(k+i)$. By using the induction hypothesis, we obtain $V(k+i+1) \subseteq V(k)$. This completes the proof. □

Now, we are ready to generalize the stability Theorem 4.3.5 for arbitrary cyclic codes. The stability theorem states that the b-symbol Hamming weight enumerator of $\mathcal{C}$ is the same for all $\dim_{\mathbb{F}_q}(\mathcal{C}) \leq b \leq n-1$.

Theorem 4.3.8 (Stability Theorem [17])
Assume that $\gcd(n,q) = 1$. *Let* $\mathcal{C}$ *be an arbitrary cyclic code of length* n *and* U *its nonzero set in* $\mathbb{Z}/n\mathbb{Z}$. *Assume that* $U \neq \{0\}$. *Let* $k = \dim_{\mathbb{F}_q} \mathcal{C}$. *For any integer* b *in the interval* $k \leq b \leq n-1$, *the* b*-symbol Hamming weight enumerator of* $\mathcal{C}$ *is the same as the* k*-symbol Hamming weight enumerator of* $\mathcal{C}$.

Proof. Let us fix the notation of this section and let $\{u_1, u_2, ..., u_\rho\}$ be a basic nonzero set of $\mathcal{C}$, meaning that $u_j \in U_j$ for each $1 \leq j \leq \rho$. Let $f(x) = a_1 x^{u_1} + a_2 x^{u_2} + \cdots + a_\rho x^{u_\rho} \in \mathbb{F}_{q^r}[x] \setminus \{0\}$ be an arbitrary nonzero polynomial in $P(\{u_1, u_2, ..., u_\rho\})$. Let $\pi_k(c(f)) \in \pi_k(\mathcal{C})$ and $\pi_b(c(f)) \in \pi_b(\mathcal{C})$ be the corresponding code words. Let $f^{(t)}(x)$ be as in definition 4.2.10. This means that

$$f^{(t)}(x) = \eta^{tu_1} a_1 x^{u_1} + \eta^{tu_2} a_2 x^{u_2} + \cdots + \eta^{tu_\rho} a_\rho x^{u_\rho}. \tag{4.13}$$

Let $c_i^{(b)}(f) \in \mathbb{F}_q^b$ be the i-th entry of the code word $\pi_b(c(f)) \in (\mathbb{F}_q^b)^n$ such that

$$\pi_b(c(f)) = \big(c_0^{(b)}(f), c_1^{(b)}(f), ..., c_{n-1}^{(b)}(f)\big).$$

Let $y_1 = a_1\eta^{iu_1}, y_2 = a_2\eta^{iu_2}, ..., y_\rho = a_\rho\eta^{iu_\rho}$. Then, we can write $c_i^{(b)}(f)$ as

$$\begin{aligned} c_i^{(b)}(f) = (&Tr(y_1 + y_2 + \cdots + y_\rho), Tr(\eta^{u_1} y_1 + \eta^{u_2} y_2 + \cdots + \eta^{u_\rho} y_\rho), ..., \\ &Tr(\eta^{(b-1)u_1} y_1 + \eta^{(b-1)u_2} y_2 + \cdots + \eta^{(b-1)u_\rho} y_\rho)). \end{aligned}$$

Similarly for the i-th symbol $c_i^{(k)}(f) \in \mathbb{F}_q^k$ of the code word $\pi_k(c(f)) \in (\mathbb{F}_q^k)^n$ we have

$$\begin{aligned} c_i^{(k)}(f) = (&Tr(y_1 + y_2 + \cdots + y_\rho), Tr(\eta^{u_1} y_1 + \eta^{u_2} y_2 + \cdots + \eta^{u_\rho} y_\rho), ..., \\ &Tr(\eta^{(k-1)u_1} y_1 + \eta^{(k-1)u_2} y_2 + \cdots + \eta^{(k-1)u_\rho} y_\rho)). \end{aligned}$$

This implies that $c_i^{(b)}(f)$ does not contribute to the Hamming weight of the code word $\pi_b(c(f))$ if and only if

$$\begin{aligned} 0 &= Tr(y_1 + y_2 + \cdots + y_\rho), \\ 0 &= Tr(\eta^{u_1} y_1 + \eta^{u_2} y_2 + \cdots + \eta^{u_\rho} y_\rho), \\ &\vdots \\ 0 &= Tr(\eta^{(b-1)u_1} y_1 + \eta^{(b-1)u_2} y_2 + \cdots + \eta^{(b-1)u_\rho} y_\rho). \end{aligned} \tag{4.14}$$

Similarly, $c_i^{(k)}(f)$ does not contribute to the Hamming weight of the code word $\pi_k(c(f))$ if

and only if

$$\begin{aligned} 0 &= Tr(y_1 + y_2 + \cdots + y_\rho), \\ 0 &= Tr(\eta^{u_1} y_1 + \eta^{u_2} y_2 + \cdots + \eta^{u_\rho} y_\rho), \\ &\vdots \\ 0 &= Tr(\eta^{(k-1)u_1} y_1 + \eta^{(k-1)u_2} y_2 + \cdots + \eta^{(k-1)u_\rho} y_\rho). \end{aligned} \tag{4.15}$$

We will prove that the conditions in (4.14) and (4.15) are equivalent. As $k \leq b$, it is clear that (4.14) implies (4.15). Therefore, assume that (4.15) is true. Let t be any integer such that $k \leq t \leq b-1$. Recall that

$$\mathbf{v_t} = [\eta^{tu_1}, \eta^{tu_2}, ..., \eta^{tu_\rho}].$$

Using Lemma 4.3.7, we have that $\mathbf{v_t} \in V(t)$ and therefore one can find $e_0, e_1, ..., e_{k-1} \in \mathbb{F}_q$ such that

$$\begin{aligned} \mathbf{v_t} &= [\eta^{tu_1}, \eta^{tu_2}, ..., \eta^{tu_\rho}], \\ &= e_0[1, 1, ..., 1] + e_1[\eta^{u_1}, \eta^{u_2}, ..., \eta^{u_\rho}] + \cdots \\ &\quad + e_{k-1}[\eta^{(k-1)u_1}, \eta^{(k-1)u_2}, ..., \eta^{(k-1)u_\rho}]. \end{aligned}$$

Using the Euclidean inner product in $\mathbb{F}_q^\rho$ on $\mathbf{v_t}$ and $[y_1, ..., y_\rho]$, we obtain

$$\begin{aligned} \langle \mathbf{v_t}, [y_1, ..., y_\rho] \rangle &= \eta^{tu_1} y_1 + \eta^{tu_2} y_2 + \cdots + \eta^{tu_\rho} y_\rho, \\ &= e_0(y_1 + y_2 + \cdots + y_\rho) + e_1(\eta^{u_1} y_1 + \eta^{u_2} y_2 + \cdots + \eta^{u_\rho} y_\rho) + \cdots \\ &\quad + e_{k-1}(\eta^{(k-1)u_1} y_1 + \eta^{(k-1)u_2} y_2 + \cdots + \eta^{(k-1)u_\rho} y_\rho). \end{aligned}$$

Taking the trace of both sides and using its $\mathbb{F}_q$-linearity we get

$$\begin{aligned} Tr(\langle \mathbf{v_t}, [y_1, ..., y_\rho] \rangle) =& Tr(\eta^{tu_1} y_1 + \eta^{tu_2} y_2 + \cdots + \eta^{tu_\rho} y_\rho), \\ =& e_0 Tr(y_1 + y_2 + \cdots + y_\rho) \\ &+ e_1 Tr(\eta^{u_1} y_1 + \eta^{u_2} y_2 + \cdots + \eta^{u_\rho} y_\rho) \\ &\vdots \\ &+ e_{k-1} Tr(\eta^{(k-1)u_1} y_1 + \eta^{(k-1)u_2} y_2 + \cdots + \eta^{(k-1)u_\rho} y_\rho). \end{aligned} \tag{4.16}$$

Since (4.15) is true by assumption, we have

$$\begin{aligned} 0 &= Tr(y_1 + y_2 + \cdots + y_\rho), \\ 0 &= Tr(\eta^{u_1} y_1 + \eta^{u_2} y_2 + \cdots + \eta^{u_\rho} y_\rho), \\ &\vdots \\ 0 &= Tr(\eta^{(k-1)u_1} y_1 + \eta^{(k-1)u_2} y_2 + \cdots + \eta^{(k-1)u_\rho} y_\rho). \end{aligned}$$

This implies that in (4.16) we have

$$Tr(\eta^{tu_1} y_1 + \eta^{tu_2} y_2 + \cdots + \eta^{tu_\rho} y_\rho) = 0.$$

This is true for each integer t where $k \leq t \leq b-1$. We conclude that the conditions in (4.14) and (4.15) are equivalent.
Recall that $\pi_b(c(f)) = (c_0^{(b)}(f), c_1^{(b)}(f), ..., c_{n-1}^{(b)}(f))$. Therefore, $c_i^{(b)}(f)$ contributes 0 or 1 to the Hamming weight of $\pi_b(c(f))$ depending on condition (4.14). Similarly $c_i^{(k)}$ contributes 0 or 1 to the Hamming weight of $\pi_k(c(f))$ depending on condition (4.15). Since both conditions are equivalent we must have $w_H(\pi_b(c(f))) = w_H(\pi_k(c(f)))$. □

Notice that the only assumption we made was that $\gcd(n, q) = 1$ and that $n \mid q^r - 1$. This gives us a large class of cyclic codes that satisfies Theorem 4.3.8. For those codes, studying the b-symbol weight distribution for $1 \leq b \leq k$ is enough. Theorem 4.3.8 gives us a strong statement on the stability of the b-symbol weight distribution. Still, it remains the question if one could determine the exact b-symbol weight distribution of any given cyclic code. In the next sections, we will see that under some assumptions, the b-symbol weight distribution can be determined entirely up to a new invariant, which we will denote as $\mu(b)$ resp. $\mu_{(i)}(b)$.

4.4 b-Symbol Weight Distribution of Semiprimitive Irreducible Cyclic Codes

The goal of this section is to completely determine the b-symbol weight distribution of semiprimitive irreducible cyclic codes. First, we make some assumptions for our convenience, and later, we will try to generalize the results. We define the integer u as

$$u := \gcd\left(\frac{q^r - 1}{q - 1}, N\right),$$

where N is a positive integer such that for a cyclic code of length n, we have $N \cdot n = q^r - 1$.

4.4.1 b-Symbol Weight Distribution of some Irreducible Cyclic Codes with $u = 2$

Let $\mathcal{C}$ be an irreducible cyclic code of length n over $\mathbb{F}_q$ and $\dim_{\mathbb{F}_q}(\mathcal{C}) = r$. Then we know from Theorem 4.3.6 that

$$\mathcal{C} = \{(Tr(a\gamma^0), Tr(a\gamma^1), ..., Tr(a\gamma^{n-1})) \ : \ a \in \mathbb{F}_{q^r}\},$$

where γ is a n-th root of unity in $\mathbb{F}_{q^r}$ and r is the smallest integer such that $q^r \equiv 1 \mod n$. For this section, we will make the following assumptions.

- $q = p^l$, where $l > 0$ is an integer and $p \neq 2$ is a prime.
- $r \geq 2$ is an even integer.
- $n \cdot N = q^r - 1$, where n, the length of the code $\mathcal{C}$, and N are positive integers.
- $\gcd(\frac{q^r-1}{q-1}, N) = 2$.
- $\eta \in \mathbb{F}_{q^r}$ is a primitive element of $\mathbb{F}_{q^r}$.

First, we want to make some remarks on the assumptions to clarify things. By the assumptions, we have that η^N is a n-th root of unity. The fact that r is an even positive integer implies that all elements in $\{0, 1, 2, ..., q-1\}$ are squares in $\mathbb{F}_{q^r}$. Furthermore, since $p \neq 2$ and r is even, we have that $\frac{q^r-1}{q-1}$ is even. The integer $\gcd(\frac{q^r-1}{q-1}, N)$ is a component of the definition 2.3.24 of semiprimitive irreducible cyclic codes and will be needed in later proofs. We fixed it to be 2 for the moment, but later, we will try to generalize it more. We define for $a \in \mathbb{F}_{q^r}$ the code word $c(a)$ as

$$c(a) = (Tr(a\eta^{0\cdot N}), Tr(a\eta^{1\cdot N}), ..., Tr(a\eta^{(n-1)\cdot N})).$$

Therefore, we can write $\mathcal{C} = \{c(a) \ : \ a \in \mathbb{F}_{q^r}\}$.
As we stated earlier, we will need a new invariant $\mu(b)$ to determine the complete b-symbol weight distribution of a cyclic code. Furthermore, to define $\mu(b)$, we need to introduce a subset of $\mathbb{F}_{q^r}^*$. Notice that $\dim_{\mathbb{F}_q}(\mathcal{C}) = r \geq b$, therefore, by Corollary 4.3.4 the set $\{1, \eta^N, \eta^{2N}, ..., \eta^{(b-1)N}\}$ is linearly independent.

Definition 4.4.1 ($\mathcal{P}(b)$ [23])
Let $2 \leq b \leq r$ be an integer. Let $\mathcal{P}(b)$ be the subset of cardinality $\frac{q^b-1}{q-1}$ in $\mathbb{F}_{q^r}^*$ defined as

$$\mathcal{P}(b) = \bigcup_{j=1}^{b-1} \{\eta^{(j-1)N} + x_1\eta^{jN} + \cdots + x_{b-j}\eta^{(b-1)N} \ : \ x_1, ..., x_{b-j} \in \mathbb{F}_q\} \cup \{\eta^{(b-1)N}\}.$$

Example 4.4.2
Let $r \geq 3$, then we have

$$\mathcal{P}(2) = \{1 + x_1\eta^N \ : \ x_1 \in \mathbb{F}_q\} \cup \{\eta^N\},$$
$$\mathcal{P}(3) = \{1 + x_1\eta^N + x_2\eta^{2N} \ : \ x_1, x_2 \in \mathbb{F}_q\} \cup \{\eta^N + x_1\eta^{2N} \ : \ x_1 \in \mathbb{F}_q\} \cup \{\eta^{2N}\}.$$

We are ready to define the invariant $\mu(b)$.

Definition 4.4.3 ($\mu(b)$ [23])
For $2 \leq b \leq r$, we define the invariant $\mu(b)$ as

$$\mu(b) = |\{x \in \mathcal{P}(b) \ : \ x \text{ is a square in } \mathbb{F}_{q^r}^*\}|$$

It is still an open problem to determine the exact value of $\mu(b)$ for $2 \leq b < r$. In the case where $b = r$, it is possible to determine $\mu(b)$ exactly, which we will see in the following lemma.

Lemma 4.4.4 (Exact Value of $\mu(r)$ [23])

$$\mu(r) = \frac{1}{2}|\mathcal{P}(b)| = \frac{q^r - 1}{2(q-1)}$$

Proof. First, notice that since r is even, every element in $\{0, 1, 2, ..., q-1\}$ is a square in $\mathbb{F}_{q^r}$. From definition 4.4.1 we obtain that

$$\mathbb{F}_{q^r}^* = \bigsqcup_{x \in \mathcal{P}(b)} x\mathbb{F}_q^*.$$

This implies that $x \in \mathcal{P}(b)$ is a square in $\mathbb{F}_{q^r}^*$ if and only if every element in $x\mathbb{F}_q^*$ is a square in $\mathbb{F}_{q^r}^*$. Since the number of squares in $\mathbb{F}_{q^r}^*$ is $\frac{q^r-1}{2}$, we obtain the equation

$$\mu(b)(q-1) = \frac{q^r-1}{2}.$$

□

Since we have defined $\mu(b)$, our next step will be to determine the b-symbol weight distribution of $\mathcal{C}$. Recall that the b-symbol weight distribution is given by the polynomial

$$A^{(b)}(T) = 1 + A_1^{(b)}T + A_2^{(b)}T^2 + \cdots + A_n^{(b)}T^n,$$

where $A_i^{(b)}$ denotes the number of code words with b-symbol Hamming weight i. The following lemmas will be useful to determine $A^{(b)}(T)$.

Lemma 4.4.5 ([23])
Let H be the multiplicative subgroup of order $\frac{q^r-1}{N}$. Consider $\mathbb{F}_q^* \times H$ as a group with componentwise multiplicative operations. Let ϕ be the map defined as

$$\begin{aligned} \phi : \mathbb{F}_q^* \times H &\to \mathbb{F}_{q^r}^* \\ (y, z) &\mapsto yz \end{aligned}$$

Then ϕ is a group homomorphism and $\ker(\phi) = \{(h, h^{-1}) \ : \ h \in H_0\}$, where $H_0 \subseteq \mathbb{F}_{q^r}^*$ is the multiplicative subgroup of order $\frac{2(q-1)}{N}$.

Proof. Note that $(y, z) \in \ker(\phi)$ if and only if $yz = 1$. Therefore, if $(y, z) \in \ker(\phi)$, then $y \in \mathbb{F}_q^* \cap H$ and $z \in \mathbb{F}_q^* \cap H$ with $z = y^{-1}$. Conversely if $h \in \mathbb{F}_q^* \cap H$, then $(h, h^{-1}) \in \ker(\phi)$. Now, we need to show that $\mathbb{F}_q^* \cap H = H_0$. This is equivalent to the statement

$$\gcd\left(q-1, \frac{q^r-1}{N}\right) = \frac{2(q-1)}{N}. \tag{4.17}$$

Note that we assumed for this section that $\gcd(\frac{q^r-1}{q-1}, N) = 2$. This gives us the equation

$$\gcd\left(\frac{q^r-1}{2(q-1)}, \frac{N}{2}\right) = 1. \tag{4.18}$$

Using the fact that $\frac{N}{2} \Big| \frac{q^r-1}{2}$ and $\frac{q^r-1}{2} = \frac{q^r-1}{2(q-1)} \cdot (q-1)$, it follows from equation (4.18) that

$$\frac{N}{2} \Big| (q-1) \tag{4.19}$$

Further, since r is even and q is odd, we have that $\frac{N}{2} \Big| \frac{q^r-1}{q-1}$. Using (4.19), we obtain

$$\frac{2(q-1)}{N} \Big| \gcd\left(q-1, \frac{q^r-1}{N}\right). \tag{4.20}$$

We have that

$$\frac{2(q-1)}{N} \cdot \frac{N}{2} = (q-1), \quad \text{and} \quad \frac{2(q-1)}{N} \cdot \frac{q^r-1}{2(q-1)} = \frac{q^r-1}{N}.$$

Therefore by (4.18) we must have

$$\gcd\left(q-1, \frac{q^r-1}{N}\right) = \frac{2(q-1)}{N}.$$

□

For the next lemma, we use the multiset notation $\{*\cdots*\}$ with the multiplicities denoted as multiplication coefficients.
E.g, the set $\{*x_1, x_2, x_2, x_2, x_3, x_3*\}$ becomes $\{*1 \cdot x_1, 3 \cdot x_2, 2 \cdot x_3*\}$.

Lemma 4.4.6 ([23])
Let $a \in \mathbb{F}_{q^r}^*$. Then

- *If a is a square, then:*

$$\{*ayz \ : \ y \in \mathbb{F}_q^*, \ z \in H*\} = \left\{*\frac{2(q-1)}{N} \cdot x \ : \ x \in \mathbb{F}_{q^r}^*, \ x \ \text{ is a square } *\right\}.$$

- *If a is a non-square, then*

$$\{*ayz \ : \ y \in \mathbb{F}_q^*, \ z \in H*\} = \left\{*\frac{2(q-1)}{N} \cdot x \ : \ x \in \mathbb{F}_{q^r}^*, \ x \ \text{ is a non-square } *\right\}.$$

Proof. Assume that $a = 1$. The homomorphism $\phi : \mathbb{F}_q^* \times H \to \mathbb{F}_{q^r}^*$ defined in Lemma 4.4.5 has a kernel of cardinality $\frac{2(q-1)}{N}$. Thus, the image of ϕ has the cardinality

$$|Im(\phi)| = \frac{(q-1)\frac{q^r-1}{N}}{\frac{2(q-1)}{N}} = \frac{q^r-1}{2}.$$

Moreover, since r is even we have that $y \in \mathbb{F}_q^*$ is a square in $\mathbb{F}_{q^r}^*$, and further because N is even too, we have that $z \in H$ is also a square in $\mathbb{F}_{q^r}^*$. This implies that $\phi(y, z)$ is a square in $\mathbb{F}_{q^r}^*$. Combining this fact with the cardinality of $Im(\phi)$, we conclude that

$$Im(\phi) = \{x \in \mathbb{F}_{q^r}^* \ : \ x \ \text{ is a square } \}.$$

Using the fact that $|\ker(\phi)| = \frac{2(q-1)}{N}$, we also get the multiplicities, which completes the proof for $a = 1$.
If $a \in \mathbb{F}_{q^r}^* \setminus \{1\}$ is a square, then the multiset $\{*ayz \ : \ y \in \mathbb{F}_q^*, z \in H\}$ is the same as the multiset $\{*yz \ : \ y \in \mathbb{F}_q^*, z \in H\}$.
If $a \in \mathbb{F}_{q^r}^*$ is a non-square, then the set $\{ayz \ : \ y \in \mathbb{F}_q^*, z \in H\}$ is the same as the set $\{x \ : \ x \text{ is a non-square } \}$. Moreover, the multiplicities are the same, which completes the proof. □

For the proof of the explicit formula of $w_b(c(a))$, we will use exponential sums and additive characters, which are important tools in number theory. For a more profound view of this topic, one can read more in [[12], Chapter 5]. Using these exponential sums, we can write the b-symbol weight of a code word as the sum of Hamming weights of multiple code words. We see this fact in the following theorem.

Theorem 4.4.7 (b-Symbol Weight as Sum of Hamming Weights [23])
Let $\mathcal{C}$ be an (n, r) irreducible cyclic code. Let $N = \frac{q^r-1}{n}$ such that η^N is a n-th root of unity in $\mathbb{F}_{q^r}$. Therefore, we can write

$$\mathcal{C} = \{c(a) = (Tr(a\eta^{0N}), Tr(a\eta^{1N}), ..., Tr(a\eta^{(n-1)N})) \ : \ a \in \mathbb{F}_{q^r}\}.$$

Let $\mathbb{F}_{q^b} = \bigcup_{\theta_i \in \mathcal{P}(b)} \theta_i \mathbb{F}_q^* \cup \{0\}$. I.e., each θ_i is an element of exactly one one-dimensional subspace of $\mathbb{F}_{q^b}$. Then, we have

$$w_b(c(a)) = \frac{1}{q^{b-1}} \sum_{i=1}^{(q^b-1)/(q-1)} w_H(c(\theta_i a)). \tag{4.21}$$

Proof. Let $I = \{1, \eta^1, ..., \eta^{n-1}\}$ and $x \in I$. Let χ be an additive character of $\mathbb{F}_{q^r}$. For $\alpha \in \mathbb{F}_q$ an additive character has the property

$$\sum_{y \in \mathbb{F}_q} \chi(y\alpha) = \begin{cases} q & \text{if } \alpha = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Using χ, we can write the Hamming weight of the code word $c(a)$ as

$$n - w_H(c(a)) = \sum_{x \in I} \frac{1}{q} \sum_{y \in \mathbb{F}_q} \chi(yax^N).$$

To obtain the b-symbol weight of $c(a)$, we need to count how many b consecutive symbols $(Tr(ax), Tr(a\eta x), ..., Tr(a\eta^b x))$ are all zero. Let $V = span\{1, \eta, ..., \eta^{b-1}\}$. We obtain

$$\begin{aligned} n - w_b(c(a)) &= \sum_{x \in I} \frac{1}{q^b} \sum_{y_1, ..., y_b \in \mathbb{F}_q} \chi(y_1 + y_2\eta + \cdots + y_{b-1}\eta^{b-1})ax^N), \\ &= \frac{1}{q^{b-1}} \sum_{x \in I} \frac{1}{q} \sum_{u \in V} \chi(uax^N). \end{aligned}$$

Considering $\mathbb{F}_{q^b} = \bigcup_{\theta_i \in \mathcal{P}(b)} \theta_i \mathbb{F}_q^* \cup 0$, we obtain

$$\begin{aligned} n - w_b(c(a)) &= \frac{1}{q^{b-1}} \sum_{i=1}^{\frac{q^b-1}{q-1}} \sum_{x \in I} \frac{1}{q} \sum_{y \in \mathbb{F}_q^*} \chi(a\theta_i a x^N) + \frac{n}{q^b}, \\ &= \frac{1}{q^{b-1}} \sum_{i=1}^{\frac{q^b-1}{q-1}} (n - w_H(c(\theta_i a))) - \frac{n(q^b-1)}{q^b(q-1)} + \frac{n}{q^b}, \\ &= \frac{n(q^b-1)}{q^{b-1}(q-1)} - \frac{1}{q^{b-1}} \sum_{i=1}^{\frac{q^b-1}{q-1}} w_H(c(\theta_i a)) - \frac{n(q^b-q)}{q^b(q-1)}, \\ &= n - \frac{1}{q^{b-1}} \sum_{i=1}^{\frac{q^b-1}{q-1}} w_H(c(\theta_i a)). \end{aligned}$$

This completes the proof. □

We are ready to state the formula for the b-symbol Hamming weight $w_b(c(a))$. Recall that $2 \leq b < r$, and $q = p^l$, where p is odd and the characteristic of $\mathbb{F}_q$.

Theorem 4.4.8 (Explicit Formula for $\mu_b(c(a))$ [23])
Let $a \in \mathbb{F}_{q^r}^*$. Assume that $2 \leq b < r$. Let $\mathcal{K}$ be the constant defined by

$$\mathcal{K} = \frac{q^b - 1}{N(q-1)q^{b-1}}$$

Then we determine $w_b(c(a))$ explicitly as follows:

- *If $p \equiv 1 \mod 4$ and a is a square in $\mathbb{F}_{q^r}^*$, then*

$$w_b(c(a)) = \mathcal{K} \cdot \left(q^r - \frac{q^r + (q-1)q^{r/2}}{q} \right) + \frac{2\mu(b)(q-1)q^{r/2}}{Nq^b}.$$

- *If $p \equiv 1 \mod 4$ and a is a non-square in $\mathbb{F}_{q^r}^*$, then*

$$w_b(c(a)) = \mathcal{K} \cdot \left(q^r - \frac{q^r - (q-1)q^{r/2}}{q} \right) - \frac{2\mu(b)(q-1)q^{r/2}}{Nq^b}.$$

- *If $p \equiv 3 \mod 4$ and a is a square in $\mathbb{F}_{q^r}^*$, then*

$$w_b(c(a)) = \mathcal{K} \cdot \left(q^r - \frac{q^r + (-1)^{lr/2}(q-1)q^{r/2}}{q} \right) + \frac{2\mu(b)(-1)^{lr/2}(q-1)q^{r/2}}{Nq^b}.$$

- *If $p \equiv 3 \mod 4$ and a is a non-square in $\mathbb{F}_{q^r}^*$, then*

$$w_b(c(a)) = \mathcal{K} \cdot \left(q^r - \frac{q^r - (-1)^{lr/2}(q-1)q^{r/2}}{q} \right) - \frac{2\mu(b)(-1)^{lr/2}(q-1)q^{r/2}}{Nq^b}.$$

Proof. For $a \in \mathbb{F}_{q^r}$, we define the extended code word $\hat{c}(a) \in \mathbb{F}_q^{q^r-1}$ as

$$\hat{c}(a) = \big(Tr(a\eta^{0\cdot N}), Tr(a\eta^{1\cdot N}), ..., Tr(a\eta^{j\cdot N}), ..., Tr(a\eta^{(q^r-1)\cdot N})\big).$$

Notice that $c(a)$ can be viewed as the shortening of $\hat{c}(a)$ to the first $n = (q^r-1)/N$ terms as we've seen in (4.7). Therefore, we obtain similar as in (4.8) that

$$w_H(c(a)) = \frac{1}{N} w(\hat{c}(a)) \quad \text{and} \quad w_b(c(a)) = \frac{1}{N} w_b(\hat{c}(a)) \tag{4.22}$$

for each $2 \leq b \leq n-1$. For the following of the proof, we will use an exponential sum $Z(a) \in \mathbb{C}$ and use some of its properties. we define $Z(a)$ as

$$Z(a) = \frac{1}{q} \sum_{y \in \mathbb{F}_q} \sum_{x \in \mathbb{F}_{q^r}} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q/p}(yTr_{q^r/q}(ax^N))}. \tag{4.23}$$

For each $x \in \mathbb{F}_{q^r}^*$, we have that

$$\sum_{y \in \mathbb{F}_q} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q/p}(yTr_{q^r/q}(ax^N))} = \begin{cases} q & \text{if } Tr_{q^r/q}(ax^N) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, using $Z(a)$ we can write the Hamming weight of $\hat{c}(a)$ as

$$w_H(\hat{c}(a)) = q^r - Z(a). \tag{4.24}$$

Let $H \subseteq \mathbb{F}_{q^r}^*$ be the multiplicative subgroup of order $\frac{q^r-1}{N}$ as we've seen in Lemma 4.4.5 and Lemma 4.4.6. We can write $Z(a)$ as

$$\begin{aligned}
Z(a) &= \frac{1}{q} \sum_{y \in \mathbb{F}_q} \sum_{x \in \mathbb{F}_{q^r}} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q/p}(yTr_{q^r/q}(ax^N))}, \\
&= \frac{1}{q} \left[q^r + \sum_{y \in \mathbb{F}_q^*} \sum_{x \in \mathbb{F}_{q^r}} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q/p}(yTr_{q^r/q}(ax^N))} \right], \\
&= \frac{1}{q} \left[q^r + \sum_{y \in \mathbb{F}_q^*} \left(1 + \sum_{x \in \mathbb{F}_{q^r}^*} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q^r/p}(ayx^N)} \right) \right], \\
&= \frac{1}{q} \left[q^r + q - 1 + \sum_{y \in \mathbb{F}_q^*} \sum_{x \in \mathbb{F}_{q^r}^*} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q^r/p}(ayx^N)} \right], \\
&= \frac{1}{q} \left[q^r + q - 1 + N \sum_{y \in \mathbb{F}_q^*} \sum_{z \in H} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q^r/p}(ayz)} \right].
\end{aligned} \tag{4.25}$$

It is a useful fact, that the sum $\sum_{y \in \mathbb{F}_{q^r}^*} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q^r/p}(y)}$ can be explicitly calculated using [[12], Theorem 5.15]. In this case, we have

$$\sum_{y \in \mathbb{F}_{q^r}^* \,:\, y \text{ is a square}} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q^r/p}(y)} = \begin{cases} \frac{-q^{r/2}-1}{2} & \text{if } p \equiv 1 \mod 4, \\ \frac{-(-1)^{lr/2}q^{r/2}-1}{2} & \text{if } p \equiv 3 \mod 4, \end{cases} \tag{4.26}$$

and

$$\sum_{y \in \mathbb{F}_{q^r}^* \,:\, y \text{ is a non-square}} e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q^r/p}(y)} = \begin{cases} \frac{q^{r/2}-1}{2} & \text{if } p \equiv 1 \mod 4, \\ \frac{(-1)^{lr/2}q^{r/2}-1}{2} & \text{if } p \equiv 3 \mod 4. \end{cases} \tag{4.27}$$

Assume that $a \in \mathbb{F}_{q^r}^*$ is a square. Using Lemma 4.4.6 and combining (4.25) and (4.26) we

obtain that

$$Z(a) = \frac{1}{q}\left[q^r + q - 1 + N\frac{2(q-1)}{N} \sum_{\substack{y\in\mathbb{F}_{q^r}^* \\ y \text{ is a square}}} e^{\frac{2\pi\sqrt{-1}}{p}Tr_{q^r/p(y)}}\right],$$

$$= \begin{cases} \frac{1}{q}\left(q^r + q - 1 + 2(q-1)\frac{-q^{r/2}-1}{2}\right) & \text{if } p \equiv 1 \mod 4, \\ \frac{1}{q}\left(q^r + q - 1 + 2(q-1)\frac{q^{r/2}-1}{2}\right) & \text{if } p \equiv 1 \mod 4, \end{cases} \tag{4.28}$$

$$= \begin{cases} \frac{q^r-(q-1)q^{r/2}}{q} & \text{if } p \equiv 1 \mod 4, \\ \frac{q^r-(-1)^{lr/2}(q-1)q^{r/2}}{q} & \text{if } p \equiv 3 \mod 4, \end{cases}$$

Similarly by taking a non-square $a \in \mathbb{F}_{q^r}^*$ and by using (4.25), (4.27), and Lemma 4.4.6 we obtain that

$$Z(a) = \begin{cases} \frac{q^r+(q-1)q^{r/2}}{q} & \text{if } p \equiv 1 \mod 4, \\ \frac{q^r+(-1)^{lr/2}(q-1)q^{r/2}}{q} & \text{if } p \equiv 3 \mod 4, \end{cases} \tag{4.29}$$

Recall that from Theorem 4.4.7, we have that

$$w_b(\hat{c}(a)) = \frac{1}{q^{b-1}} \sum_{\theta\in\mathcal{P}(b)} w_H(\hat{c}(\theta a)). \tag{4.30}$$

Combining (4.24) and (4.30) we obtain

$$w_b(\hat{c}(a)) = \frac{1}{q^{b-1}} \sum_{\theta\in\mathcal{P}(b)} \big(q^r - Z(\theta a)\big), \tag{4.31}$$

where we can see the importance of knowing which elements of $\mathcal{P}(b)$ are squares resp. non-squares. Recall that we denote the number of squares in $\mathcal{P}(b)$ as $\mu(b)$. The rest of this proof consists of going through all cases where a is a square resp. non-square and $p \equiv 1 \mod 4$ resp. $p \equiv 3 \mod 4$.
Assume $p \equiv 1 \mod 4$ and a is a square in $\mathbb{F}_{q^r}^*$. Combining (4.28), (4.29) and (4.31) we obtain that

$$\begin{aligned} w_b(\hat{c}(a)) &= \frac{1}{q^{b-1}}\left(\mu(b)\left[q^r - \frac{q^r-(q-1)q^{r/2}}{q}\right]\right) \\ &+ \frac{1}{q^{b-1}}\left(\left(\frac{q^b-1}{q-1} - \mu(b)\right)\left[q^r - \frac{q^r+(q-1)q^{r/2}}{q}\right]\right) \\ &= \frac{q^b-1}{(q-1)q^{b-1}}\left(q^r - \frac{q^r+(q-1)q^{r/2}}{q}\right) + \frac{2\mu(b)(q-1)q^{r/2}}{q^b}. \end{aligned} \tag{4.32}$$

Analog we obtain for $p \equiv 1 \mod 4$ and a non-square in $\mathbb{F}_{q^r}^*$ that

$$w_b(\hat{c}(a)) = \frac{q^b-1}{(q-1)q^{b-1}}\left(q^r - \frac{q^r-(q-1)q^{r/2}}{q}\right) - \frac{2\mu(b)(q-1)q^{r/2}}{q^b}. \tag{4.33}$$

For $p \equiv 3 \mod 4$ and a square in $\mathbb{F}_{q^r}^*$ we obtain

$$w_b(\hat{c}(a)) = \frac{q^b-1}{(q-1)q^{b-1}}\left(q^r - \frac{q^r+(-1)^{lr/2}(q-1)q^{r/2}}{q}\right) + \frac{2\mu(b)(-1)^{lr/2}(q-1)q^{r/2}}{q^b}. \tag{4.34}$$

For $p \equiv 3 \mod 4$ and a non-square in $\mathbb{F}_{q^r}^*$ we obtain

$$w_b(\hat{c}(a)) = \frac{q^b-1}{(q-1)q^{b-1}}\left(q^r - \frac{q^r-(-1)^{lr/2}(q-1)q^{r/2}}{q}\right) - \frac{2\mu(b)(-1)^{lr/2}(q-1)q^{r/2}}{q^b}. \tag{4.35}$$

Combining (4.32), (4.33), (4.34), (4.35) with (4.22) we complete the proof. □

Now, we have an explicit form of $w_b(c(a))$ for $2 \leq b < r$. Recall that the stability theorem (Theorem 4.3.8) states that $w_b(c(a)) = w_r(c(a))$ for all $r \leq b \leq n-1$. Using the same techniques as in the proof of Theorem 4.4.8 for $b = r$, we obtain the following theorem.

Theorem 4.4.9 ([23])
Let $a \in \mathbb{F}_{q^r}^*$. For $r \leq b < n$ we have

$$w_b(c(a)) = n.$$

Proof. If $r \leq b \leq n-1$ then we know by the stability theorem (Theorem 4.3.8) that $w_b(c(a)) = w_r(c(a))$. Therefore, it is enough to determine $w_r(c(a))$.
Assume $p \equiv 1 \mod 4$ and a is a square in $\mathbb{F}_{q^r}^*$. Lemma 4.4.4 implies that $\mu(b) = \frac{q^r-1}{2(q-1)}$. If we apply Theorem 4.4.8 we get

$$\begin{aligned} w_r(c(a)) &= \frac{q^r-1}{N(q-1)q^{r-1}}\left(q^r - \frac{q^r+(q-1)q^{r/2}}{q}\right) + \frac{(q^r-1)q^{r/2}}{Nq^r} \\ &= \frac{q^r-1}{N(q-1)q^{r-1}}(q^r - q^{r-1}) + \frac{q^r-1}{Nq^r}(-q^{r/2}+q^{r/2}) \\ &= \frac{(q^r-1)q^{r-1}}{Nq^{r-1}} = \frac{q^r-1}{N} = n. \end{aligned}$$

The proof for $p \equiv 1 \mod 4$ and a non-square is similar.
Assume now that $p \equiv 3 \mod 4$ and a is a square in $\mathbb{F}_{q^r}^*$. Then again by applying Theorem 4.4.8 we obtain that

$$\begin{aligned} w_r(c(a)) &= \frac{q^r-1}{N(q-1)q^{r-1}}\left(q^r - \frac{q^r+(-1)^{lr/2}(q-1)q^{r/2}}{q}\right) + \frac{(-1)^{lr/2}(q^r-1)q^{r/2}}{Nq^r} \\ &= \frac{q^r-1}{N(q-1)q^{r-1}}(q^r - q^{r-1}) + \frac{(-1)^{lr/2}(q^r-1)}{Nq^r}(-q^{r/2}+q^{r/2}) \\ &= \frac{(q^r-1)q^{r-1}}{Nq^{r-1}} = \frac{q^r-1}{N} = n. \end{aligned}$$

The proof for $p \equiv 3 \mod 4$ and a non-square is similar. □

We can combine Theorem 4.4.8 and Theorem 4.4.9 to completely determine the Hamming weight enumerator of $\pi_b(\mathcal{C})$.

Corollary 4.4.10 (b-Symbol Hamming Weight Enumerator of $\mathcal{C}$ [23])
For $2 \leq b \leq r-1$, the b-symbol Hamming weight enumerator of $\mathcal{C}$ is

$$A^{(b)}(T) = 1 + \frac{q^r - 1}{2}(T^{u_1} + T^{u_2}),$$

where

$$u_1 = \begin{cases} \frac{q^b-1}{N(q-1)q^{b-1}}\left(q^r - \frac{q^r+(q-1)q^{r/2}}{q}\right) + \frac{2\mu(b)(q-1)q^{r/2}}{Nq^b} & \text{if } p \equiv 1 \mod 4, \\ \frac{q^b-1}{N(q-1)q^{b-1}}\left(q^r - \frac{q^r+(-1)^{lr/2}(q-1)q^{r/2}}{q}\right) + \frac{2\mu(b)(-1)^{lr/2}(q-1)q^{r/2}}{Nq^b} & \text{if } p \equiv 3 \mod 4. \end{cases}$$

and

$$u_2 = \begin{cases} \frac{q^b-1}{N(q-1)q^{b-1}}\left(q^r - \frac{q^r-(q-1)q^{r/2}}{q}\right) - \frac{2\mu(b)(q-1)q^{r/2}}{Nq^b} & \text{if } p \equiv 1 \mod 4, \\ \frac{q^b-1}{N(q-1)q^{b-1}}\left(q^r - \frac{q^r-(-1)^{lr/2}(q-1)q^{r/2}}{q}\right) - \frac{2\mu(b)(-1)^{lr/2}(q-1)q^{r/2}}{Nq^b} & \text{if } p \equiv 3 \mod 4. \end{cases}$$

For $r \leq b \leq n-1$, the b-symbol Hamming weight enumerator of $\mathcal{C}$ is

$$A^{(b)}(T) = 1 + (q^r - 1)T^n.$$

Recall that in definition 3.3.10, we defined *MDS* b-symbol codes as b-symbol codes of length n, size M, and minimum b-distance d_b such that M attains the Singleton bound, $M = q^{n-d_b+b}$. We can observe that for $b = r$, $\mathcal{C}$ is a b-symbol MDS code, since we have

$$q^r = q^{n-n+r} = q^r.$$

Corollary 4.4.11
Let $b = r$. Then $\mathcal{C}$ is a b-symbol MDS code.

Recall that we assumed until now that $u = \gcd((q^r-1)/(q-1), N) = 2$. The next step is, therefore, to consider some arbitrary u.

4.4.2 b-Symbol Weight Distribution of Semiprimitive Irreducible Cyclic Codes with arbitrary u

For this part, we keep the assumptions from the previous part with the change that we don't fix $u = 2$. We set

$$u = \gcd\left(\frac{q^r - 1}{q - 1}, N\right)$$

Next, we define cyclotomic classes of order u.

Definition 4.4.12 (Cyclotomic Classes of Order u)
We define the cyclotomic classes of order u in $\mathbb{F}_{q^r}$ as

$$C_i^{(u,q^r)} := \eta^i \langle \eta^u \rangle$$

for $i = 0, 1, ..., u-1$. Here, $\langle \eta^u \rangle$ denotes the subgroup of $\mathbb{F}_{q^r}^*$ generated by η^u.

It will also be helpful to recall the notion of semiprimitive irreducible cyclic codes.

Definition 4.4.13 (Semiprimitive Irreducible Cyclic Codes [21])
Let $\gcd(p,u)=1$, *p is semiprimitive modulo u if there exists a positive integer d such that* $u \mid (p^d+1)$. *An (n,r) irreducible cyclic code over $\mathbb{F}_q$ is semiprimitive if $u \geq 2$ and p is semiprimitive modulo u. Almost every irreducible code with $u \geq 2$ is semiprimitive.*

In fact, one can check in [16] that excluding subfield codes, there exist only eleven irreducible codes with $u \geq 2$, which are not semiprimitive. Notice that the codes we considered in the last section where we fixed $u=2$ are semiprimitive irreducible cyclic codes. Our goal is to find the b-symbol enumerator of all semiprimitive irreducible cyclic codes with $u \geq 1$. As in (4.23) we want to use the properties of the additive character

$$\chi(x) := e^{\frac{2\pi\sqrt{-1}}{p} Tr_{q^r/q}(x)} \quad \text{for } x \in \mathbb{F}_{q^r}.$$

Recall that for $a \in \mathbb{F}_{q^r}$ we have

$$\sum_{x \in \mathbb{F}_{q^r}} \chi(ax) = \begin{cases} q & \text{if } a=0, \\ 0 & \text{otherwise.} \end{cases}$$

The main idea is to split $\chi(x)$ in sums over the cyclotomic classes. We write

$$\gamma_i^{(u,q^r)} := \sum_{x \in C_i^{(u,q^r)}} \chi(x).$$

As in (4.23), we can explicitly calculate these sums. See for instance [14] or ([12], Chapter 5).

Theorem 4.4.14 ([21])
Suppose that $rl = 2sd$ and $u \mid (p^d+1)$ for some positive integers s, d and l. Then we have

$$\gamma_i^{(u,q^r)} = \begin{cases} \frac{(-1)^{s-1}(u-1)q^{r/2}-1}{u} & \text{if } i \equiv \delta \mod u, \\ \frac{(-1)^s q^{r/2}-1}{u} & \text{if } i \not\equiv \delta \mod u, \end{cases}$$

where the integer δ is defined as

$$\delta := \begin{cases} 0 & \text{if } u=1, \text{ or } p=2, \text{ or } p>2 \text{ with } 2 \mid s, \text{ or } p>2 \text{ with } 2 \nmid s, \text{ and } 2 \mid \frac{p^d+1}{u}, \\ \frac{u}{2} & \text{if } p>2, 2 \nmid s \text{ with } 2 \nmid \frac{p^d+1}{u}. \end{cases}$$

Similarly to Lemma 4.4.6, we obtain the following lemma.

Lemma 4.4.15 ([21])
For $0 \leq i < N$, we have the following equality of multisets:

$$\left\{ *xy \ : \ y \in \mathbb{F}_{q^r}^*, \ x \in C_i^{(N,q^r)} * \right\} = \left\{ * \frac{(q-1)u}{N} x \ : \ x \in C_i^{(u,q^r)} * \right\}$$

In fact, the subgroup H we defined in 4.4.5 equals $C_0^{(N,q^r)}$. Recall that for $1 \leq b \leq r$ we defined the set $\mathcal{P}(b)$ as

$$\mathcal{P}(b) := \bigcup_{j=1}^{b-1} \{\eta^{(j-1)N} + x_1\eta^{jN} + \cdots + x_{b-j}\eta^{(b-1)N} \ : \ x_1, ..., x_{b-j} \in \mathbb{F}_q\} \cup \{\eta^{(b-1)N}\},$$

and we set $\mu(b) = |\{x \in \mathcal{P}(b) \ : \ x \text{ is a square in } \mathbb{F}_{q^r}\}|$. Note that we can also write

$$\mu(b) = |\{x \in \mathcal{P}(b) \ : \ x \in C_0^{(2,q^r)}\}|.$$

This gives us the following generalization of $\mu(b)$.

Definition 4.4.16 (Generalization of the μ-Invariant)
Let $1 \leq b \leq r$. For $0 \leq i < u$, we define

$$\mu_{(i)}(b) := |\{x \in \mathcal{P}(b) \ : \ x \in C_i^{(u,q^r)}\}|.$$

We will use $\mu_{(i)}(b)$ to determine the b-weight distribution from Corollary 4.4.10 for all semiprimitive irreducible cyclic codes. Recall that we have already proved everything for $u = 2$. Therefore, the proofs for general $u = \gcd((q^r-1)/(q-1), N)$ will be very similar. First, we generalize the results from Lemma 4.4.4.

Lemma 4.4.17 (Exact Value of $\mu_{(i)}(r)$ [21])
If $b = r$ then, for any $0 \leq i < u$ we have

$$\mu_{(i)}(r) = \frac{|\mathcal{P}(b)|}{u} = \frac{q^r-1}{u(q-1)}.$$

Proof. The proof is very similar to the proof of Lemma 4.4.4. We have that

$$\mathbb{F}_{q^r}^* = \bigsqcup_{x \in \mathcal{P}(b)} x\mathbb{F}_q^*.$$

Since $u \mid (q^r-1)/(q-1)$ and $\langle \eta^{(q^r-1)/(q-1)} \rangle = \mathbb{F}_q^*$, we have $x \in C_i^{(u,q^r)}$ if and only if all elements of $x\mathbb{F}_q^*$ are also in $C_i^{(u,q^r)}$, which has $(q^r-1)/u$ elements. This gives us the equation

$$\mu_{(i)}(r)(q-1) = \frac{q^r-1}{u}.$$

□

Now, we will determine the Hamming weight distribution of semiprimitive irreducible cyclic codes with $u \geq 1$. We will use this result to finally determine the b-symbol weight distribution of semiprimitive irreducible cyclic codes. Recall that $\mathcal{C} = \{c(a) \ : \ a \in \mathbb{F}_{q^r}\}$ where

$$c(a) = (Tr(a\eta^{0N}), Tr(a\eta^{1N}), ..., Tr(a\eta^{(n-1)N})).$$

Theorem 4.4.18 (Hamming Weight Distribution of Semiprimitive Codes [21])
Let $u = \gcd((q^r-1)/(q-1), N)$. Assume that $u = 1$ or p is semiprimitive modulo u. Let d be the smallest positive integer such that $u \mid (p^d+1)$ and let $s = 1$ if $u = 1$ and $s = (rl)/(2d)$ if $u > 1$. We set

$$W_A = \frac{nq^{r/2-1}(q-1)}{q^r-1}(q^{r/2} - (-1)^{s-1}(u-1)),$$
$$W_B = \frac{nq^{r/2-1}(q-1)}{q^r-1}(q^{r/2} - (-1)^s).$$

Then, $\mathcal{C}$ is an (n, r) irreducible cyclic code with the following Hamming weight enumerator:

$$A(T) = 1 + \frac{q^r-1}{u}T^{W_A} + \frac{(q^r-1)(u-1)}{u}T^{W_B}. \tag{4.36}$$

Proof. Similarly to the proof of Theorem 4.4.8 we can write

$$\begin{aligned} w_H(c(a)) &= n - \frac{1}{q}\sum_{i=0}^{n-1}\sum_{y\in\mathbb{F}_q} e^{\frac{2\pi\sqrt{-1}}{p}Tr_{q/p}(yTr_{q^r/q}(a\eta^{Ni}))} \\ &= n - \frac{n}{q} - \frac{1}{q}\sum_{y\in\mathbb{F}_{q^r}^*}\sum_{x\in C_0^{(N,q^r)}}\chi(yax) \\ &= n - \frac{n}{q} - \frac{(q-1)u}{qN}\sum_{z\in C_0^{(u,q^r)}}\chi(az), \end{aligned}$$

where the last equality follows from Lemma 4.4.15. Suppose that $a \in C_i^{(u,q^r)}$ for some $0 \le i < u$. Then we have

$$\begin{aligned} w_H(c(a)) &= n - \frac{n}{q} - \frac{(q-1)}{qN}u\gamma_i^{(u,q^r)} \\ &= \frac{n(q-1)}{q(q^r-1)}(q^r-1) - \frac{n(q-1)}{q(q^r-1)}u\gamma_i^{(u,q^r)} \\ &= \frac{nq^{r-1}(q-1)}{q^r-1} - \frac{n(q-1)}{q(q^r-1)}(u\gamma_i^{(u,q^r)}+1) \\ &= \frac{nq^{r-1}(q-1)}{q^r-1} - \frac{nq^{r/2-1}(q-1)}{q^r-1}\cdot\frac{(u\eta_i^{(u,q^r)}+1)}{q^{r/2}} \\ &= \frac{nq^{r/2-1}(q-1)}{q^r-1}\left(q^{r/2} - \frac{u\gamma_i^{(u,q^r)}+1}{q^{r/2}}\right). \end{aligned}$$

Let δ be as in Theorem 4.4.14. We have that

$$i \equiv \delta \mod u \iff a \in C_\delta^{(u,q^r)}.$$

Therefore, by Theorem 4.4.14 we have that

$$w_H(c(a)) = \begin{cases} W_A & \text{if } a \in C_\delta^{(u,q^r)}, \\ W_B & \text{if } a \in \mathbb{F}_{q^r}^* \setminus C_\delta^{(u,q^r)}. \end{cases} \tag{4.37}$$

The proof follows now from the fact that $|C_\delta^{(u,q^r)}| = \frac{q^r-1}{u}$ and $|\mathbb{F}_{q^r}^* \setminus C_\delta^{(u,q^r)}| = \frac{(q^r-1)(u-1)}{u}$. □

We are now ready to determine the b-symbol weight distribution of semiprimitive irreducible cyclic codes.

Theorem 4.4.19 (b-Symbol Weight Distribution of Semiprimitive Codes [21]) *Let $\mathcal{P}(b)$, $\mu_{(i)}(b)$ and δ be as before. Then, for $0 \le i < u$ and $1 \le b \le r$, we have*

$$W_i^{(b)} = \frac{(q-1)q^{r/2-b}}{N}\Big(|\mathcal{P}(b)|(q^{r/2}-(-1)^s) + (-1)^s u\mu_{(\delta-i \mod u)}(b)\Big). \tag{4.38}$$

Then, the b-symbol Hamming weight enumerator of $\mathcal{C}$ is given by

$$A^{(b)}(T) = 1 + \frac{q^r-1}{u}\sum_{i=0}^{u-1} T^{W_i^{(b)}}. \tag{4.39}$$

Proof. Let W_A and W_B be as in Theorem 4.4.18 and suppose that $a \in C_i^{(u,q^r)}$ for some $0 \leq i < u$. Using equation (4.37), we have that $w_H(c(\theta a)) = W_A$ if and only if $\theta a \in C_\delta^{(u,q^r)}$ if and only if $\theta \in C_{(\delta-i \mod u)}^{(u,q^r)}$. There are exactly $\mu_{(\delta-i \mod u)}(b)$ elements $\theta \in \mathcal{P}(b)$ satisfying $\theta \in C_{(\delta-i \mod u)}^{(u,q^r)}$. Recall that from Theorem 4.4.7, we know that

$$w_b(c(a)) = \frac{1}{q^{b-1}} \sum_{\theta \in \mathcal{P}(b)} w_H(c(\theta a)).$$

This implies that $w_b(c(a)) = W_i^{(b)}$ where

$$W_i^{(b)} = \frac{1}{q^{b-1}} \Big(\mu_{(\delta-i \mod u)}(b) W_A + (|\mathcal{P}(b)| - \mu_{(\delta-i \mod u)}(b)) W_B \Big).$$

By using the explicit values of W_A and W_B and by considering $|C_i^{(u,q^r)}| = \frac{q^r-1}{u}$, we obtain the b-symbol Hamming weight enumerator. □

Notice that Theorem 4.4.19 is also true for $b = 1$. In this case, the 1-symbol weight enumerator equals the Hamming weight enumerator from Theorem 4.4.18. Therefore, Theorem 4.4.19 generalizes Theorem 4.4.8 (the $u = 2$ case) and Theorem 4.4.18. We will now look at some examples.

Example 4.4.20 ([21])

1. *Let* $(q, r, N, b) = (2, 4, 3, 2)$. *This gives us* $u = \gcd((2^4-1)/(2-1), 3) = \gcd(15, 3) = 3$, $s = 2$, $\delta = 0$, *and* $|\mathcal{P}(2)| = q + 1 = 3$.
   Let η *be the root of* $x^4+x+1 \in \mathbb{F}_2[x]$, *which implies that* $\mathbb{F}_{16} = \mathbb{F}_2(\eta)$. *This implies that* $\mathcal{P}(2) = \{\eta^0, \eta^3, 1+\eta^3 = \eta^{14}\}$. *This gives us* $\mu_{(0)}(2) = 2$, $\mu_{(1)}(2) = 0$, *and* $\mu_{(2)}(2) = 1$. *Using Theorem 4.4.18 and Theorem 4.4.19 we obtain*

   $$W_A = 4,\ W_B = 2,\ W_0^{(2)} = 5,\ W_1^{(2)} = 4,\ W_2^{(2)} = 3.$$

   Therefore, the 2*-symbol Hamming weight enumerator of* $\mathcal{C}$ *is*

   $$A^{(2)}(T) = 1 + 5(T^3 + T^4 + T^5).$$

2. *Let* $(q, r, N, b) = (4, 3, 9, 2)$. *This gives us* $u = \gcd((4^3-1)/(4-1), 9) = \gcd(21, 9) = 3$, $s = 3$, $\delta = 0$, *and* $|\mathcal{P}(2)| = 4 + 1 = 5$.
   Let $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ *with* $\alpha^2 + \alpha + 1 = 0$. *Let* η *be the zero of* $x^3 + x^2 + x + \alpha \in \mathbb{F}_4[x]$. *This implies that* $\mathbb{F}_{64} = \mathbb{F}_4(\eta)$. *This gives us* $\mathcal{P}(2) = \{\eta^0, \eta^9, 1+\eta^9, 1+\alpha\eta^9, 1+\alpha^2\eta^9\} = \{\eta^0, \eta^9, \eta^{27}, \eta^5, \eta^{40}\}$. *This implies that* $\mu_{(0)}(2) = 3$, $\mu_{(1)}(2) = 1$, *and* $\mu_{(2)}(2) = 1$. *Therefore, by applying Theorem 4.4.18 and 4.4.19 we obtain*

   $$W_A = 4,\ W_B = 6,\ W_0^{(2)} = 6,\ W_1^{(2)} = 7,\ W_2^{(2)} = 7.$$

   Therefore, the 2*-symbol Hamming weight enumerator of* $\mathcal{C}$ *is*

   $$A^{(2)}(T) = 1 + 21(T^6 + 2T^7).$$

3. *Let* $(q, r, N, b) = (5, 5, 4, 3)$. *This gives us* $u = \gcd((5^5-1)/(5-1), 4) = \gcd(781, 4) = 1$, *and* $|\mathcal{P}(3)| = 5^2 + 5 + 1 = 31$.
Notice that since $u = 1$, *we need only* $\mu_{(0)}(3) = |\mathcal{P}(3)| = 31$. *We have*

$$W_A = 625,\ W_B = 625,\ W_0^{(3)} = 775.$$

Therefore, the 3*-symbol Hamming weight enumerator of* $\mathcal{C}$ *is*

$$A^{(3)}(T) = 1 + 3124T^{775}.$$

From example 4.4.20 (3.), we know that for the case $u = 1$ it is really easy to compute the b-symbol weight enumerator.

Theorem 4.4.21 ([21])
Let $u = \gcd((q^r - 1)/(q - 1), N) = 1$, *then for any* $1 \leq b \leq r$, *the* b*-symbol Hamming weight enumerator of* $\mathcal{C}$ *is given by*

$$A^{(b)}(T) = 1 + (q^r - 1)T^{\frac{q^r - q^{r-b}}{N}}$$

Furthermore, analog to Theorem 4.4.9 we obtain the following theorem.

Theorem 4.4.22 ([21])
Let $a \in \mathbb{F}_{q^r}^*$. *For* $r \leq b < n$ *we have*

$$w_b(c(a)) = n$$

Proof. Using the stability theorem (Theorem 4.3.8), it is enough to compute $w_r(c(a))$. Suppose $a \in C_i^{(u,q^r)}$ for some $0 \leq i < u$. Therefore, by the proof of Theorem 4.4.19 we have that $w_r(c(a)) = W_i^{(r)}$ where

$$W_i^{(r)} = \frac{(q-1)q^{r/2-r}}{N}\Big(|\mathcal{P}(r)|(q^{r/2} - (-1)^s) + (-1)^s u \mu_{(\delta - i \mod u)}(r)\Big).$$

Using Lemma 4.4.17, we obtain $\mu_{(\delta - i \mod u)}(r) = \frac{q^r-1}{u(q-1)}$. We also have $|\mathcal{P}(r)| = \frac{q^r-1}{q-1}$ and $n = \frac{q^r-1}{N}$. By computing now $W_i^{(r)}$, we obtain $w_r(c(a)) = n$. □

Similar to Corollary 4.4.11, we have that $\mathcal{C}$ is an r-symbol MDS code.

Corollary 4.4.23
Let $b = r$. *Then* $\mathcal{C}$ *is a* b*-symbol MDS code*

We conclude that we have determined the b-symbol weight distribution of all semiprimitive irreducible cyclic codes with $u \geq 1$. However, the main problem remains. Namely, there are no formulas or good lower and upper bounds for the invariants $\mu(b)$ resp. $\mu_{(i)}(b)$. Therefore, the rest of the thesis will focus on finding the properties of these invariants. To do so, we will simplify the setting and focus on symbol-pair codes over fields of the form $\mathbb{F}_{q^4}$. We will also focus on semiprimitive irreducible cyclic codes where we fix $u = 2$. This means that we consider $\mu_{(0)}(2) = \mu(2) = |\{x \in \mathcal{P}(2) \ : \ x \text{ is a square in } \mathbb{F}_{q^4}\}|$. We will quickly deduce that $\mu(2)$ depends on the choice of the primitive element $\eta \in \mathbb{F}_{q^4}^*$ which we use to describe the code $\mathcal{C}$. This motivates us to investigate the average of all $\mu(2)$ over all primitive elements since the sum of all $\mu(2)$ over all primitive elements is a fixed integer.

Chapter 5

Properties of the Sum of μ-Invariants of Symbol-Pair Codes.

In Chapter 4, we determined the b-symbol enumerator for all semiprimitive irreducible cyclic codes. The problem with the formulas of the b-symbol enumerator from Theorem 4.4.8 or Theorem 4.4.19 is that the computation of $\mu(b)$ resp. $\mu_{(i)}(b)$ is very time-consuming since we have to check each element of $\mathcal{P}(b)$ in which cyclotomic coset $C_i^{(u,q^r)}$ it is. There is still no formula or a useful algorithm that we can use to compute $\mu(b)$ resp. $\mu_{(i)}(b)$ in a sophisticated way. In this chapter, we want to investigate these invariants and we obtain completely new results in Section 5.2 and Section 5.3.
There is also another problem regarding the set of $\mathcal{P}(b)$. In general, the number of squares in $\mathcal{P}(b)$ depends on the choice of the primitive element of $\mathbb{F}_{q^r}^*$, which implies that $\mu(b)$ resp. $\mu_{(i)}(b)$ is not fixed. We see this in the following example.

Example 5.0.1
Consider the field $\mathbb{F}_{7^4} = \mathbb{F}_7[x]/ < x^4 + x^2 + 3x + 5 >$, *where* $x^4 + x^2 + 3x + 5$ *is a minimal polynomial, i.e. an irreducible polynomial with a primitive element of* $\mathbb{F}_{7^4}$ *as its root. Let* η *be the root of* $x^4 + x^2 + 3x + 5$. *This implies that all* η^i, *where* i *and* $7^4 - 1 = 2400$ *are coprime, are primitive elements of* $\mathbb{F}_{7^4}$.
Now consider the set $\mathcal{P}(2) = \{1 + xw^{q-1} \ : \ x \in \mathbb{F}_7\} \cup \{w^{q-1}\} = \{1 + xw^6 \ : \ x \in \mathbb{F}_7\} \cup \{w^6\}$, *where* $w = \eta^i$ *and* $\gcd(i, 2400) = 1$. *Notice that* $\gcd((q^4-1)/(q-1), q-1) = \gcd(400, 6) = 2$. *This means that* $q - 1$ *is the* N *from the assumptions in section 4.4.1. Since* $7^4 - 1 = 2400$ *is not that big, one can easily count each value of* $\mu(2)$ *for each primitive element* $w \in \mathbb{F}_{7^4}^*$. *To do so we use the Matlab code from Appendix A.1, where we create* $\mathbb{F}_{7^4}$ *and* $\mathcal{P}(2)$ *for each primitive element and check for each element in* $\mathcal{P}(2)$ *whether it is in the set of squares of* $\mathbb{F}_{7^4}$ *or not. If we plot all the results in a bar chart, we obtain figure 5.1. We notice that* $\mu(2)$ *depends on the choice of the primitive element. Therefore, if* $b = 2$, *for each primitive element* $w \in \mathbb{F}_{q^r}^*$ *we will define the sets*

$$\mathcal{P}(w) = \{1 + xw^{q-1} \ : \ x \in \mathbb{F}_q\} \cup \{w^{q-1}\}$$

and,

$$\mu(w) = |\{\alpha \in \mathcal{P}(w) \ : \ \alpha \ \textit{ is a square in } \mathbb{F}_{q^r}\}|.$$

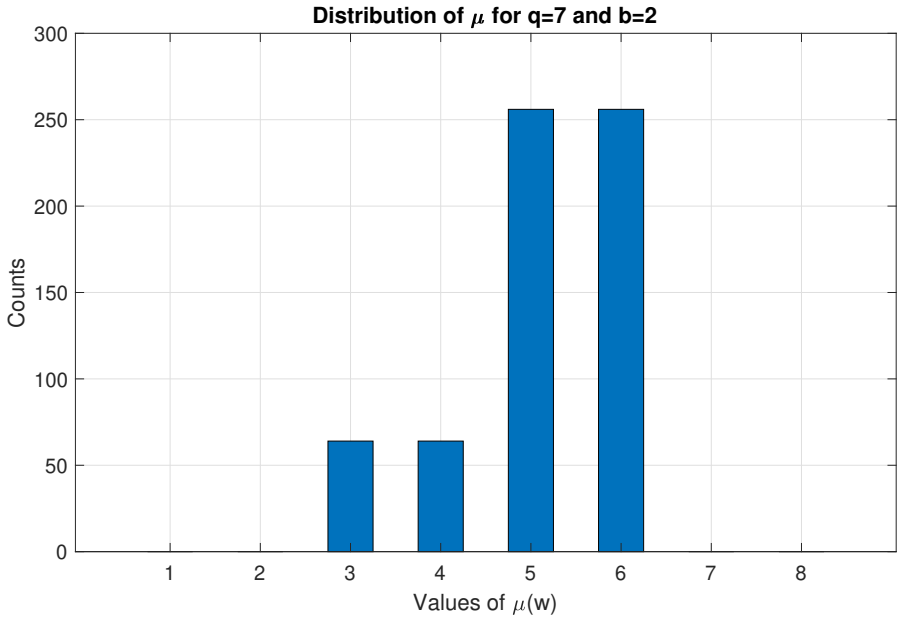

Figure 5.1: This plot describes the amount of squares in $\mathcal{P}(w)$ for each primitive element $w \in \mathbb{F}_{7^4}^*$. There are $\varphi(2400) = 640$ primitive elements. There are 64 primitive elements attaining each the value $\mu(w) = 3$ and the value $\mu(w) = 4$. There are also 256 primitive elements attaining each the value $\mu(w) = 5$ and the value $\mu(w) = 6$. The code for the calculations can be found in Appendix A.1.

For the following sections, we make the following assumptions similar to those assumptions from section 4.4.1.

- $b = 2$, (i.e., we consider symbol-pair codes).
- $q = p^l$, where $l > 0$ and $p \neq 2$ is a prime and $q \equiv 3 \mod 4$.
- $r = 4$.
- $N = q - 1$ such that $n \cdot N = q^4 - 1$, where n is the length of the code $\mathcal{C}$.
- W is the set of primitive elements of $\mathbb{F}_{q^4}^*$.
- D is the set of squares of $\mathbb{F}_{q^4}^*$.
- $\mathcal{P}(w) = \{1 + xw^{q-1} \ : \ x \in \mathbb{F}_q\} \cup \{w^{q-1}\}$.
- $\mu(w) = |\{\alpha \in \mathcal{P}(w) \ : \ \alpha \ \text{ is a square in } \mathbb{F}_{q^4}\}|$.

Notice that for $r = 2$ and $b = 2$, the invariants are given by $\mu(w) = (q^2 - 1)/(2(q - 1))$ according to Lemma 4.4.4. So the simplest case to study the distribution of non-trivial $\mu(w)$ is for $r = 4$.
Since $r = 4$, the multiplicative group $\mathbb{F}_{q^4}^*$ has $q^4 - 1$ elements. Since we want

$$\gcd\left(\frac{q^4-1}{q-1}, q-1\right) = \gcd\left(q^3+q^2+q+1, q-1\right) = 2,$$

we need that $q \equiv 3 \mod 4$. The assumptions are basically an adaption from the assumptions of section 4.4.1 for the fields of the form $\mathbb{F}_{q^4}$.
We know from example 5.0.1 that $\mu(w)$ depends on w. However, the sum of $\mu(w)$ over all $w \in W$ is a fixed positive integer. By using some function field theory and number theory, one can determine lower and upper bounds on $\sum_{w \in W} \mu(w)$.

5.1 Bounds of the Sum of μ-Invariants of Symbol-Pair Codes

In this section, we derive lower and upper bounds for $\sum_{w \in W} \mu(w)$. This part is based on [2] with the difference that we include some trivial bounds. We obtain trivial lower and upper bounds by multiplying the minimal resp. the maximal possible values of $\mu(w)$ times the number of primitive elements, which is given by the Euler phi function $\varphi(\cdot)$.

Lemma 5.1.1 (Trivial Bounds)
The sum of the μ-invariants is trivially bounded by

$$2 \cdot \varphi(q^4 - 1) \leq \sum_{w \in W} \mu(w) \leq (q+1) \cdot \varphi(q^4 - 1).$$

Proof. The upper bound follows directly from the definition of $\mathcal{P}(w)$ as a set of cardinality $q + 1$.
The lower bound follows from the fact that $w^{q-1}, 1 + 0 \cdot w^{q-1} = 1 \in \mathcal{P}(w)$ are always squares for all $w \in W$. □

Our goal in this section is to obtain a more useful bound than the trivial bounds. The main idea is to consider the equations of the form $x + ax^q \in \mathbb{F}_{q^4}[x]$, where $a \in \mathbb{F}_q$. Clearly, we have $x + ax^q \in D$ if and only if $1 + ax^{q-1} \notin D$. Especially for $a = 1$, the curve $y^2 = x + x^q$ is beneficial since it is a subcover of the Hermitian curve $y^{q+1} = x + x^q \in \mathbb{F}_{q^2}[x]$, which has useful properties. For a more profound view on examples of algebraic function fields as the Hermitian, we refer to ([18], Chapter 6).

Let $w \in W$. We start by splitting $\mu(w)$ in the following way:

$$\mu^*(w) = |\{a \in \mathbb{F}_q \setminus \{0, 1, -1\} \ : \ 1 + aw^{q-1} \in D\}|,$$

$$\mu^{(1)}(w) = \begin{cases} 1 & \text{if } 1 + w^{q-1} \in D, \\ 0 & \text{otherwise .} \end{cases}$$

$$\mu^{(-1)}(w) = \begin{cases} 1 & \text{if } 1 - w^{q-1} \in D, \\ 0 & \text{otherwise .} \end{cases}$$

Considering also the trivial squares $1, w^{q-1} \in \mathcal{P}(w)$ we obtain

$$\mu(w) = \mu^*(w) + \mu^{(1)}(w) + \mu^{(-1)}(w) + 2. \tag{5.1}$$

Analog to $\mu(w)$ we define

$$\eta^*(w) = |\{a \in \mathbb{F}_q \setminus \{0, 1, -1\} \ : \ w + aw^q \in D\}|,$$
$$\eta^{(1)}(w) = \begin{cases} 1 & \text{if } w + w^q \in D, \\ 0 & \text{otherwise .} \end{cases}$$
$$\eta^{(-1)}(w) = \begin{cases} 1 & \text{if } w - w^q \in D, \\ 0 & \text{otherwise .} \end{cases}$$

Since for all $0 \neq a \in \mathbb{F}_q$ we have either $1 + aw^{q-1} \in D$ or $w + aw^q \in D$, we have that

$$\mu^*(w) + \eta^*(w) = q - 3, \tag{5.2}$$

and

$$\mu^{(1)}(w) + \eta^{(1)}(w) = \mu^{(-1)}(w) + \eta^{(-1)}(w) = 1. \tag{5.3}$$

Let $\mathcal{A}$ be the multiset

$$\mathcal{A} = \{*x + ax^q \in D \ : \ x \in \mathbb{F}_{q^4}^*, \ a \in \mathbb{F}_q \setminus \{0, 1, -1\}*\}.$$

Let $A^{(1)}$ and $A^{(-1)}$ be the sets

$$A^{(1)} = \{x \in \mathbb{F}_{q^4}^* \ : \ x + x^q \in D\},$$
$$A^{(-1)} = \{x \in \mathbb{F}_{q^4}^* \ : \ x - x^q \in D\}.$$

Notice that for $\mathcal{A}$, $A^{(1)}$ and $A^{(-1)}$ we consider all $x \in \mathbb{F}_{q^4}^*$ satisfying the property that $x + ax^q$ is a square in $\mathbb{F}_{q^4}$ for $a \in \mathbb{F}_q \setminus \{0\}$. Our main interest is in knowing which of these $x \in \mathbb{F}_{q^4}$ is in W. So we define analog to $\mathcal{A}$ the following sets

$$\mathcal{M} = \{*x + ax^q \in D \ : \ x \in W, a \in \mathbb{F}_q \setminus \{0, 1, -1\}*\},$$
$$M^{(1)} = \{x \in W \ : \ x + x^q \in D\},$$
$$M^{(-1)} = \{x \in W \ : \ x - x^q \in D\}.$$

where $\mathcal{M}$ is a multiset. Using these sets, we obtain the following proposition.

Proposition 5.1.2 ([2])
Let $\mathcal{M}$, $M^{(1)}$ and $M^{(-1)}$ be as defined above, then we have

$$\sum_{w \in W} \mu(w) = (q+1)\varphi(q^4 - 1) - |\mathcal{M}| - |M^{(1)}| - |M^{(-1)}|.$$

Proof. Note that we can identify $\mathcal{M}$, $M^{(1)}$ and $M^{(-1)}$ as sums of η^*, $\eta^{(1)}$ and $\eta^{(-1)}$ as follows:

$$\begin{aligned} \sum_{w \in W} \eta^*(w) &= |\mathcal{M}|, \\ \sum_{w \in W} \eta^{(1)}(w) &= |M^{(1)}|, \\ \sum_{w \in W} \eta^{(-1)}(w) &= |M^{(-1)}|. \end{aligned} \tag{5.4}$$

By using (5.2) and (5.3) we have that

$$\begin{aligned}\sum_{w\in W}\Big(\mu^*(w)+\eta^*(w)+\mu^{(1)}(w)+\eta^{(-1)}(w)+\mu^{(-1)}(w)+\eta^{(-1)}(w)\Big),\\ =(q-3)|W|+|W|+|W|,\\ =(q-1)\varphi(q^4-1).\end{aligned}\tag{5.5}$$

By using (5.1), we also have that

$$\begin{aligned}\sum_{w\in W}\Big(\mu^*(w)+\mu^{(1)}(w)+\mu^{(-1)}(w)\Big),\\ =\sum_{w\in W}(\mu(w)-2),\\ =-2\varphi(q^4-1)+\sum_{w\in W}\mu(w).\end{aligned}\tag{5.6}$$

Combining (5.4), (5.5) and (5.6) we obtain

$$|\mathcal{M}|+|M^{(1)}|+|M^{(-1)}|-2\varphi(q^4-1)+\sum_{w\in W}\mu(w)=(q-1)\varphi(q^4-1),$$

which completes the proof. □

Our goal is to find useful bounds for $|\mathcal{M}|$, $|M^{(1)}|$ and $|M^{(-1)}|$. To do so, we want first to know the cardinalities of $\mathcal{A}$, $A^{(1)}$ and $A^{(-1)}$. We will find these cardinalities in the following section.

5.1.1 Cardinalities of $\mathcal{A}$, $A^{(1)}$ and $A^{(-1)}$

Proposition 5.1.3 (Cardinality of $\mathcal{A}$ [2])
Let $\mathcal{A}=\{*x+ax^q\in D\ :\ x\in\mathbb{F}_{q^4}^*,\ a\in\mathbb{F}_q\setminus\{0,1,-1\}*\}$ be as defined above. Then we have

$$|\mathcal{A}|=\frac{(q-3)(q^4-1)}{2}.\tag{5.7}$$

Proof. First, we show that for $a\in\mathbb{F}_q\setminus\{0,1,-1\}$ the map $x\mapsto x+ax^q$ is a permutation on $\mathbb{F}_{q^4}^*$. Notice that $x\mapsto x+ax^q$ on $\mathbb{F}_{q^4}$ is a $\mathbb{F}_q$-linear map. This follows from the fact that $(x+y)^q=x^q+y^q$ and $(cx)^q=cx^q$ for all $x,y\in\mathbb{F}_{q^4}$ and $c\in\mathbb{F}_q$. Therefore, it is enough to prove that the kernel of $x\mapsto x+ax^q$ is trivial.
Assume that a non-zero $x\in\mathbb{F}_{q^4}$ exists such that $x+ax^q=0$. This implies that $1+ax^{q-1}=0$. Thus, a must satisfy $a=\frac{-1}{x^{q-1}}$. Notice that $(q-1)(q^3+q^2+q+1)=q^4-1$ and so by taking the (q^3+q^2+q+1)-th power on both sides we obtain

$$a^{q^3+q^2+q+1}=\left(\frac{-1}{x^{q-1}}\right)^{q^3+q^2+q+1}=1.$$

Notice, that $q^3+q^2+q+1\equiv 4 \mod q-1$. Therefore, since $a\in\mathbb{F}_q$, a must satisfy $a^4=1$. Furthermore, since $q\equiv 3 \mod 4$, we know that $q-1\equiv 2 \mod 4$. This implies that

$\gcd(q-1,4)=2$ and therefore $a^2=1$. However, $a^2=1$ implies also that $a\in\{-1,1\}$, which contradicts the assumption that $a\in\mathbb{F}_q\setminus\{0,1,-1\}$. We conclude that the map $x\mapsto x+ax^q$ is a permutation on $\mathbb{F}_{q^4}^*$. Therefore, the set $\{x+ax^q \ : \ x\in\mathbb{F}_{q^4}^*\}=\mathbb{F}_{q^4}^*$, and similar $\{x+ax^q\in D \ : \ x\in\mathbb{F}_{q^4}^*\}=D$ for all $a\in\mathbb{F}_q\setminus\{0,1,-1\}$. This implies that

$$|\mathcal{A}|=\frac{(q-3)(q^4-1)}{2}.$$

□

It remains to find the cardinalities of $A^{(1)}$ and $A^{(-1)}$. First, we consider the cardinality of $A^{(1)}$. We will need algebraic function field theory to solve this problem. We start by introducing some useful function fields and their properties. For more detail on these function fields and the proofs of their properties, we refer to [18].

Definition 5.1.4 (Hermitian Function Field [18])
The Hermitian function field over $\mathbb{F}_{q^2}$ can be defined by

$$\mathcal{H}=\mathbb{F}_{q^2}(x,y) \quad with \quad y^q+y=x^{q+1}. \tag{5.8}$$

It has the following properties:

1. *The genus of $\mathcal{H}$ is $g=\frac{q(q-1)}{2}$.*
2. *$\mathcal{H}/\mathbb{F}_{q^2}$ is a maximal function field, i.e. the number of rational places of $\mathcal{H}$ over $\mathbb{F}_{q^2}$, denoted by $N(\mathcal{H})$, satisfies that*

$$N(\mathcal{H})=(q+1)+2gq^{2/2}=(q+1)+q^3-q=q^3+1.$$

Definition 5.1.5 (Kummer Extension [18])
Let F/K be an algebraic function field where K contains a primitive n-th root of unity, where $n>1$ and $\gcd(n,char(K))=1$. Suppose $u\in F$ satisfies $u\neq w^d$ for all $w\in F$ and $d\mid n$, $d>1$. Let

$$F'=F(y) \quad with \quad y^n=u.$$

Then we call the extension F'/F the Kummer extension of F.

We need especially Kummer extensions similar to the following example.

Example 5.1.6 ([18])
Let $F=\mathbb{F}_{q^r}(x,y)$ with $y^2=f(x)=p_1(x)p_2(x)\cdots p_s(x)\in\mathbb{F}_{q^r}[x]$, where $p_1(x),p_2(x),...,p_s(x)$ are distinct irreducible polynomials and $s\geq 1$. Let $m=\deg f(x)$. Then $\mathbb{F}_{q^r}$ is the constant field of F, and $F/\mathbb{F}_{q^r}$ has genus

$$g=\begin{cases}(m-1)/2 & \text{if } m\equiv 1 \mod 2,\\ (m-2)/2 & \text{if } m\equiv 0 \mod 2.\end{cases} \tag{5.9}$$

Let $N(F_r)$ denote the number of rational places of $F\mathbb{F}_{q^r}$. Recall that from 2.2.25, we have that

$$N(F_r)=q^r+1-\sum_{i=1}^{2g}\alpha_i^r,$$

where g is the genus of F and $\alpha_1, ..., \alpha_{2g}$ are the reciprocals of the L-polynomial $L_{F_r}(t) \in \mathbb{C}[t]$. Suppose that F is maximal, i.e. $N(F) = (q+1) + 2gq^{1/2}$. If r is even, then we must have that F_r is minimal, i.e. $N(F_r) = (q^r+1) - 2gq^{1/2}$. This follows from the fact that for an even r, the negative α_i becomes positive as α_i^r, and the positive α_i stays positive as α_i^r. This leads to the following lemma.

Lemma 5.1.7
Let $F_1 = F\mathbb{F}_{q^2}$ be a maximal function field. Then $F_2 = F\mathbb{F}_{q^4}$ is a minimal function field.

Let us return now to determine the cardinality of $A^{(1)} = \{x \in \mathbb{F}_{q^4}^* \ : \ x + x^q \in D\}$.

Proposition 5.1.8 (Cardinality of $A^{(1)}$ [2])
The cardinality of $A^{(1)}$ is given by

$$|A^{(1)}| = \frac{q(q-1)(q^2+1)}{2}. \tag{5.10}$$

Proof. consider the function field $F = \mathbb{F}_{q^2}(x,y)$ with $x + x^q = y^2$. F is a subcover of the Hermitian function field $\mathcal{H} = \mathbb{F}_{q^2}(x,y)$ with $x + x^q = y^{q+1}$. As $\mathcal{H}$ is maximal, we must have that F is maximal too. Therefore, by Lemma 5.1.7, we must have that $F_2 = F\mathbb{F}_{q^4}$ is a minimal function field. Further, from example 5.1.6 we know that the genus of F_2 is $(q-1)/2$, since $\deg(x^q + x) = q$ is odd. Therefore, we obtain

$$N(F_2) = q^4 + 1 - 2gq^{4/2} = q^4 + 1 - (q-1)q^2 = q^4 + 1 - q^3 + q^2.$$

Notice that $x + x^q$ splits over $\mathbb{F}_{q^4}$, and each of its zeros corresponds to exactly one rational place of F_2. There is also one rational place corresponding to the pole of (x). Therefore, by subtracting all $(q+1)$ rational places corresponding to the zeros resp. the poles of $x + x^q$ we obtain

$$\begin{aligned}
|A^{(1)}| &= |\{x \in \mathbb{F}_{q^4} \ : \ x + x^q \in D\}|, \\
&= \frac{1}{2}|\{x \in \mathbb{F}_{q^4} \ : \ \exists y \in \mathbb{F}_{q^4}^* \text{ such that } x + x^q = y^2\}|, \\
&= \frac{1}{2}\big((q^4 + 1 - q^3 + q^2) - (q+1)\big), \\
&= \frac{q(q-1)(q^2+1)}{2}.
\end{aligned}$$

□

It remains now to determine the cardinality of $A^{(-1)} = \{x \in \mathbb{F}_{q^4} \ : \ x - x^q \in D\}$. In fact, we will see that $|A^{(1)}| = |A^{(-1)}|$.

Proposition 5.1.9 (Cardinality of $A^{(-1)}$ [2])
The cardinality of $A^{(-1)}$ is given by

$$|A^{(-1)}| = \frac{q(q-1)(q^2+1)}{2}. \tag{5.11}$$

Proof. Consider the curve $y^2 = x - x^q$ over $\mathbb{F}_{q^4}$. Let $\eta \in \mathbb{F}_{q^4}^*$ be a primitive element. We define

$$\begin{aligned} \alpha &:= \eta^{(q^2+1)/2}, \\ \theta &:= \alpha^{q+1} \end{aligned} \tag{5.12}$$

Notice that we have

$$\theta^{q-1} = \eta^{\frac{q^2+1}{2}(q+1)(q-1)} = \eta^{\frac{q^4-1}{2}} = -1.$$

This implies that

$$\theta^q = -\theta. \tag{5.13}$$

Consider the change of variable $(x, y) \mapsto (x_1, y_1)$ with $x = \theta x_1$ and $y = \alpha y_1$. Furthermore, this change of variable transforms $\mathbb{F}_{q^4}(x)[y]/ < x + x^q - y^2 >$, which is the same as F_2 from the proof of Proposition 5.1.8, to $\mathbb{F}_{q^4}(x)[y]/ < x - x^q - y^2 >$. Indeed, we have

$$x + x^q = (\theta x_1) + (\theta x_1)^q = \theta x_1 - \theta x_1^q = \theta(x_1 - x_1^q),$$

and

$$y^{q+1} = (\alpha y_1)^{q+1} = \theta y_1^{q+1}.$$

This implies that $|A^{(-1)}| = |A^{(1)}| = \frac{q(q-1)(q^2+1)}{2}$. □

Since we have found now the cardinalities of $\mathcal{A}$, $A^{(1)}$ and $A^{(-1)}$, we will use them to obtain bounds on $|\mathcal{M}|$, $|M^{(1)}|$ and $|M^{(-1)}|$.

5.1.2 Bounds on $|\mathcal{M}|$

We want to obtain bounds on $|\mathcal{M}|$. We achieve this by investigating the complement $\mathcal{M}^c$ and determining bounds on its cardinality in $\mathcal{A}$. We define the following complement sets:

$$\begin{aligned} W^c &:= \mathbb{F}_{q^4}^* \setminus W, \\ \mathcal{M}^c &:= \{*x + ax^q \in D \ : \ x \in W^c, \ a \in \mathbb{F}_q \setminus \{0, 1, -1\}*\}, \\ M_a^c &:= \{x \in W^c \ : \ x + ax^q \in D\}, \end{aligned}$$

where $a \in \mathbb{F}_q \setminus \{0, 1, -1\}$. Note that the sets M_a^c satisfy

$$|\mathcal{M}^c| = \sum_{a \in \mathbb{F}_q \setminus \{0,1,-1\}} |M_a^c|. \tag{5.14}$$

Let S be the set of prime factors of $q^4 - 1$. For a nonempty $I \subseteq S$ let $m = \prod_{r \in I} r$. We denote the multiplicative subgroup of $\mathbb{F}_{q^4}^*$ of order $(q^4 - 1)/m$ as $G(I)$. Further, by the definition of W^c, we have that

$$W^c = \bigcup_{I \subseteq S, \ I \neq \emptyset} G(I). \tag{5.15}$$

Using this union, we can split M_a^c into the following subsets

$$M_a^c(I) := \{x \in G(I) \ : \ x + ax^q \in D\},$$

where $a \in \mathbb{F}_q \setminus \{0, 1, -1\}$. From the inclusion-exclusion principle of sets, see for instance ([1], theorem 7.1.1), we have that

$$|M_a^c| = \sum_{I \subseteq S,\ I \neq \emptyset} (-1)^{|I|+1} |M_a^c(I)|. \tag{5.16}$$

We are able to establish bounds on $|M_a^c(I)|$ by using the Hasse-Weil bounds from Theorem 2.2.27.

Proposition 5.1.10 ([2])
Let $I \subseteq S$ be a non-empty set and $m = \prod_{r \in I} r$. Let $UB_1(I)$ and $LB_1(I)$ be the integers defined in the following way:

Case 1: *m is odd:*
In this case, we have

$$UB_1(I) = \min\left\{ \frac{q^4-1}{2m} + \frac{q-1}{2}q^2, \frac{q^4-1}{m} \right\},$$
$$LB_1(I) = \max\left\{ \frac{q^4-1}{2m} - \frac{q-1}{2}q^2, 0 \right\}.$$

Case 2: *m is even:*
In this case, we have

$$UB_1(I) = \min\left\{ \left\lfloor \frac{q^4-2q^2-3}{2m} \right\rfloor + \frac{q-1}{2}q^2, \frac{q^4-1}{m} \right\},$$
$$LB_1(I) = \max\left\{ \left\lceil \frac{q^4+2q^2-3}{2m} \right\rceil - \frac{q-1}{2}q^2, 0 \right\}.$$

Then we have $LB_1(I) \leq |M_a^c(I)| \leq UB_1(I)$.

Proof. Let N be the number of elements $x \in \mathbb{F}_{q^4}^*$ such that there exists $y \in \mathbb{F}_{q^4}^*$ satisfying

$$\chi : \ y^2 = x^m + ax^{qm}.$$

Then, we have $|M_a^c(I)| = N/m$. Notice that, since $a \notin \{0, -1, 1\}$, the curve χ satisfies

$$x^m + ax^{qm} = 0 \iff x = 0.$$

From [18] we know that the genus of the curve χ is

$$g = \begin{cases} \frac{m(q-1)}{2} & \text{if } m \equiv 1 \mod 2, \\ \frac{m(q-1)}{2} - 1 & \text{if } m \equiv 0 \mod 2. \end{cases}$$

Suppose that m is odd. Using the Hasse-Weil bounds and the fact that we are only interested in half of the rational places of $\mathbb{F}_{q^4}(x)[y]/ < x^m + ax^{qm} - y^2 >$, (i.e. we count the solutions

$(x,y),(x,-y) \in (\mathbb{F}_{q^4}^*)^2$ as one count of N), and by using the ramification structure of Kummer extensions, we obtain

$$\begin{aligned}
& q^4+1-2-2gq^2 \leq 2N \leq q^4+1-2+2gq^2, \\
\iff & \frac{(q^4-1)}{2} - \frac{m(q-1)}{2}q^2 \leq N \leq \frac{(q^4-1)}{2} + \frac{m(q-1)}{2}q^2, \\
\iff & \frac{(q^4-1)}{2m} - \frac{q-1}{2}q^2 \leq \frac{N}{m} \leq \frac{(q^4-1)}{2m} + \frac{q-1}{2}q^2.
\end{aligned}$$

Considering also the trivial upper bound $|G(I)| = \frac{q^4-1}{m}$, and the trivial lower bound 0, we obtain

$$LB_1(I) \leq |M_a^c(I)| \leq UB_1(I).$$

Now, suppose that m is even. Using the same arguments as in the m is odd case, we obtain

$$\begin{aligned}
& q^4+1-4-2gq^2 \leq 2N \leq q^4+1-4+2gq^2, \\
\iff & \frac{(q^4-3)}{2} - gq^2 \leq N \leq \frac{(q^4-3)}{2} + gq^2, \\
\iff & \frac{(q^4-3)}{2} - \frac{m(q-1)}{2}q^2 + q^2 \leq N \leq \frac{(q^4-3)}{2} + \frac{m(q-1)}{2}q^2 - q^2, \\
\iff & \left\lceil \frac{q^4+2q^2-3}{2m} \right\rceil - \frac{q-1}{2}q^2 \leq \frac{N}{m} \leq \left\lfloor \frac{q^4-2q^2-3}{2m} \right\rfloor + \frac{q-1}{2}q^2.
\end{aligned}$$

Considering also the trivial upper bound $|G(I)| = \frac{q^4-1}{m}$, and the trivial lower bound 0, we obtain

$$LB_1(I) \leq |M_a^c(I)| \leq UB_1(I).$$

□

The main idea in obtaining bounds on $|\mathcal{M}^c|$ is to use the inclusion-exclusion principle in (5.16). For instance, we have the following upper resp. lower bound

$$\sum_{I \subsetneq S,\ |I|:\ \text{odd}} |M_a^c(I)| \ - \sum_{I \subseteq S,\ |I|:\ \text{even}} |M_a^c(I)| \leq |M_a^c|$$

and

$$|M_a^c| \leq \sum_{I \subseteq S,\ |I|:\ \text{odd}} |M_a^c(I)| \ - \sum_{I \subsetneq S,\ |I|:\ \text{even}} |M_a^c(I)|.$$

Then, by replacing the $|M_a^c(I)|$ by the bounds obtained in Proposition 5.1.10, we obtain upper and lower bounds for $|M_a^c|$. This leads to the following upper resp. lower bound:

$$\sum_{\substack{I \subseteq S, \\ |I|:\ \text{odd}}} LB_1(I) \ - \sum_{\substack{I \subseteq S, \\ |I|:\text{even}}} UB_1(I) \leq |M_a^c| \leq \sum_{\substack{I \subseteq S, \\ |I|:\ \text{odd}}} UB_1(I) \ - \sum_{\substack{I \subseteq S, \\ |I|:\text{even}}} LB_1(I).$$

Notice that we don't have to consider all $I \subseteq S$ to obtain bounds on $|M_a^c|$. We want to choose $I \subseteq S$ so that we can optimize the upper and lower bound.

First, we consider the upper bound of $|M_a^c|$. Let $2 \leq t_{\text{even}} \leq |S|$ be an even integer. We define $R(t_{\text{even}})$ as the sum

$$R(t_{\text{even}}) = \sum_{\substack{t_{\text{even}}+1 \leq t \leq |S| \\ t\text{: odd}}} \sum_{I \subseteq S,\ |I|=t} UB_1(I) \ - \sum_{\substack{t_{\text{even}} \leq t \leq |S| \\ t\text{: even}}} \sum_{I \subseteq S,\ |I|=t} LB_1(I).$$

We only want to consider the $I \subseteq S$ such that $R(t_{\text{even}}) > 0$. Let I_{even} be the set of all t_{even} satisfying $R(t_{\text{even}}) > 0$. We define

$$t^*_{\text{even}} = \min\{t_{\text{even}} \in I_{\text{even}} \ : \ R(t_{\text{even}}) \geq R(t) \ \text{ for all } t \in I_{\text{even}}\}.$$

In other words, t^*_{even} is the smallest integer in I_{even} such that $R(t^*_{\text{even}}) \geq R(t)$ for all $t \in I_{\text{even}}$. The idea of this definition is that we only have to consider the $I \subseteq S$ with $1 \leq |I| \leq t^*_{\text{even}} - 1$ to obtain the upper bound of $|M_a^c|$. In the case where $I_{\text{even}} = \emptyset$, we have to consider all $I \subseteq S$.

Definition 5.1.11 (Upper Bound of $|M_a^c|$)
Let I_{even} and t^*_{even} be as discussed above. Then, we define the upper bound of $|M_a^c|$ as follows:

Case 1: *$I_{even} = \emptyset$*
In this case we have to consider all $I \subseteq S$ and obtain

$$UB_1 = \sum_{\substack{1 \leq t \leq |S| \\ t\text{: odd}}} \sum_{I \subseteq S,\ |I|=t} UB_1(I) \ - \sum_{\substack{2 \leq t \leq |S| \\ t\text{: even}}} \sum_{I \subseteq S,\ |I|=t} LB_1(I).$$

Case 2: *$I_{even} \neq \emptyset$*
In this case we only consider $I \subseteq S$ with $|I| \leq t^*_{even} - 1$ and obtain

$$UB_1 = \sum_{\substack{1 \leq t \leq t^*_{even}-1 \\ t\text{: odd}}} \sum_{I \subseteq S,\ |I|=t} UB_1(I) \ - \sum_{\substack{2 \leq t \leq t^*_{even}-2 \\ t\text{: even}}} \sum_{I \subseteq S,\ |I|=t} LB_1(I).$$

The construction of the lower bound is similar to the construction of the upper bound. Assume that $|S| \geq 3$. Let $3 \leq t_{\text{odd}} \leq |S|$ be an odd integer. Analog to $R(t_{\text{even}})$ we define $L(t_{\text{odd}})$ as

$$L(t_{\text{odd}}) = \sum_{\substack{t_{\text{odd}} \leq t \leq |S| \\ t\text{: odd}}} \sum_{I \subseteq S,\ |I|=t} LB_1(I) \ - \sum_{\substack{t_{\text{odd}}+1 \leq t \leq |S| \\ t\text{: even}}} \sum_{I \subseteq S,\ |I|=t} UB_1(I).$$

Let I_{odd} be the set containing all odd integers $3 \leq t_{\text{odd}} \leq |S|$ with $L(t_{\text{odd}}) < 0$. Similar to t^*_{even} we define the $t^*_{\text{odd}} \in I_{\text{odd}}$ as

$$t^*_{\text{odd}} = \min\{t_{\text{odd}} \in I_{\text{odd}} \ : \ L(t_{\text{odd}}) \leq L(t) \ \text{ for all } t \in I_{\text{odd}}\}$$

We only want to consider $I \subseteq S$ with $|I| \leq t^*_{\text{odd}}$. If $I_{\text{odd}} \neq 0$ or $|S| < 3$, we have to consider all $I \subseteq S$.

Definition 5.1.12 (Lower Bound of $|M_a^c|$)
Let I_{odd} and t^*_{odd} be as discussed above. Then, we define the lower bound of $|M_a^c|$ as follows:

Case 1: *$I_{odd} = \emptyset$ or $|S| < 3$:*
In this case we have to consider all $I \subseteq S$ and obtain

$$LB_1 = \sum_{\substack{1 \le t \le |S| \\ t:\ odd}} \sum_{I \subseteq S,\ |I|=t} LB_1(I) \ - \sum_{\substack{2 \le t \le |S| \\ t:\ even}} \sum_{I \subseteq S,\ |I|=t} UB_1(I).$$

Case 2: *$I_{odd} \neq \emptyset$:*
In this case we only consider $I \subseteq S$ with $|I| \le t^*_{odd} - 1$ and obtain

$$LB_1 = \sum_{\substack{1 \le t \le t^*_{odd}-2 \\ t:\ odd}} \sum_{I \subseteq S,\ |I|=t} LB_1(I) \ - \sum_{\substack{2 \le t \le t^*_{odd}-1 \\ t:\ even}} \sum_{I \subseteq S,\ |I|=t} UB_1(I).$$

Therefore, by construction of LB_1 and UB_1, we obtain the following proposition.

Proposition 5.1.13 (Bounds of $|M_a^c|$ [2])
Let LB_1 and UB_1 be as defined above. We have

$$LB_1 \le |M_a^c| \le UB_1.$$

Essentially, this implies that $(q-3)LB_1 \le |\mathcal{M}^c| \le (q-3)UB_1$, which is equivalent to

$$\begin{aligned} (q-3)LB_1 \le |\mathcal{A}| - |\mathcal{M}| \le (q-3)UB_1, \\ -|\mathcal{A}| + (q-3)LB_1 \le -|\mathcal{M}| \le -|\mathcal{A}| + (q-3)UB_1. \end{aligned} \tag{5.17}$$

5.1.3 Bounds on $|M^{(1)}|$ and $|M^{(-1)}|$

In this section, we proceed similarly to the previous one. We determine the bounds of $|M^{(1)}|$ and $|M^{(-1)}|$ by determining bounds on their complements. We denote the complements by

$$M^{(1),c} = \{x \in W^c \ : \ x + x^q \in D\},$$

and

$$M^{(-1),c} = \{x \in W^c \ : \ x - x^q \in D\}.$$

Let S be the set of prime factors of $q^4 - 1$. Let $I \subseteq S$ and $m = \prod_{r \in I} r$. We define the subsets of $M^{(1),c}$ and $M^{(-1),c}$ corresponding to $G(I)$ as

$$M^{(1),c}(I) = \{x \in G(I) \ : \ x + x^q \in D\},$$

and

$$M^{(-1),c}(I) = \{x \in G(I) \ : \ x - x^q \in D\}.$$

Using the inclusion-exclusion principle as in (5.16), we obtain

$$|M^{(1),c}| = \sum_{I \subseteq S} (-1)^{|I|+1} |M^{(1),c}(I)|, \ \text{ and } |M^{(-1),c}| = \sum_{I \subseteq S} (-1)^{|I|+1} |M^{(-1),c}(I)|.$$

Analog to Proposition 5.1.10, we obtain the following proposition on the bounds of $|M^{(1),c}(I)|$ and $|M^{(-1),c}(I)|$.

Proposition 5.1.14 ([2])
Let $I \subseteq S$ be a non-empty set and $m = \prod_{r \in I} r$. Let $UB_2(I)$ and $LB_2(I)$ be the integers defined in the following way:

Case 1: *m is odd:*
In this case we have

$$UB_2(I) = \min\left\{\frac{q^4-1}{2m} + \frac{q-1}{2}q^2 - \frac{1}{2}\gcd\left(q-1, \frac{q^4-1}{m}\right), \frac{q^4-1}{m}\right\},$$
$$LB_2(I) = \max\left\{\frac{q^4-1}{2m} - \frac{q-1}{2}q^2 - \frac{1}{2}\gcd\left(q-1, \frac{q^4-1}{m}\right), 0\right\}.$$

Case 2: *m is even:*
In this case we have

$$UB_2(I) = \min\left\{\left\lfloor\frac{q^4-2q^2-3}{2m}\right\rfloor + \frac{q-1}{2}q^2 - \frac{1}{2}\gcd\left(q-1, \frac{q^4-1}{m}\right), \frac{q^4-1}{m}\right\},$$
$$LB_2(I) = \max\left\{\left\lceil\frac{q^4+2q^2-3}{2m}\right\rceil - \frac{q-1}{2}q^2 - \frac{1}{2}\gcd\left(q-1, \frac{q^4-1}{m}\right), 0\right\}.$$

Then we have

$$LB_2(I) \leq |M^{(i),c}(I)| \leq UB_2(I), \quad \text{for } i \in \{1,-1\}.$$

Proof. The proof is the same as in Proposition 5.1.10, where we considered the curve $y^2 = x^m + ax^{mq}$. By setting $a = 1$ resp. $a = -1$, we can proceed analog to the proof of Proposition 5.1.10. The only issue is that there are $x \in \mathbb{F}_{q^4}^*$ that satisfy both equations $x^m + x^{mq} = 0$ and $x^m - x^{mq} = 0$. The number of those elements is given by $(q-1)\gcd(m, (q^4-1)/(q-1))$. □

Moreover, we can deduce lower and upper bounds on $|M^{(1),c}|$ the same way as with $|M_a^c|$. We have only to replace $UB_1(I)$ with $UB_2(I)$ and $LB_1(I)$ with $LB_2(I)$. For deducing an upper bound of $|M^{(1),c}|$, we consider for even integers $2 \leq t_{\text{even}} \leq |S|$

$$R^{(1)}(t_{\text{even}}) = \sum_{\substack{t_{\text{even}}+1 \leq t \leq |S| \\ t\text{: odd}}} \sum_{I \subseteq S,\, |I|=t} UB_2(I) \; - \sum_{\substack{t_{\text{even}} \leq t \leq |S| \\ t\text{: even}}} \sum_{I \subseteq S,\, |I|=t} LB_2(I).$$

Then we define $I_{\text{even}}^{(1)}$ as the set containing all even integers $2 \leq t_{\text{even}} \leq |S|$ such that $R^{(1)}(t_{\text{even}}) > 0$. We define $t_{\text{even}}^{(1),*}$ as the smallest integer in $I_{\text{even}}^{(1)}$ such that $R^{(1)}(t_{\text{even}}^{(1),*}) \geq R^{(1)}(t)$ for all $t \in I_{\text{even}}^{(1)}$. Essentially, we replace I_{even}, t_{even}^* from definition 5.1.11 with $I_{\text{even}}^{(1)}$ and $t_{\text{even}}^{(1),*}$ to obtain an upper bound for $|M^{(1),c}|$ resp. an upper bound for $|M^{(-1),c}|$.

Definition 5.1.15 (Upper Bound of $|M^{(1),c}|$ resp. $|M^{(-1),c}|$)
Let $I_{even}^{(1)}$ and $t_{even}^{(1),*}$ be as discussed above. Then, we define the upper bound of $|M^{(1),c}|$ resp. $|M^{(-1),c}|$ as follows:

Case 1: *$I_{even}^{(1)} = \emptyset$*
In this case we have to consider all $I \subseteq S$ and obtain

$$UB_2 = \sum_{\substack{1 \leq t \leq |S| \\ t\text{: odd}}} \sum_{I \subseteq S,\, |I|=t} UB_2(I) \; - \sum_{\substack{2 \leq t \leq |S| \\ t\text{: even}}} \sum_{I \subseteq S,\, |I|=t} LB_2(I).$$

Case 2: $I^{(1)}_{even} \neq \emptyset$
In this case we only consider $I \subseteq S$ *with* $|I| \leq t^{(1),*}_{even} - 1$ *and obtain*

$$UB_2 = \sum_{\substack{1 \leq t \leq t^{(1),*}_{even}-1 \\ t:\ odd}} \sum_{I \subseteq S,\ |I|=t} UB_2(I) \ - \sum_{\substack{2 \leq t \leq t^{(1),*}_{even}-2 \\ t:\ even}} \sum_{I \subseteq S,\ |I|=t} LB_2(I).$$

Analog, we deduce lower bounds on $|M^{(1),c}|$ resp. on $|M^{(-1),c}|$ by defining

$$L^{(1)}(t_{\text{odd}}) = \sum_{\substack{t_{\text{odd}} \leq t \leq |S| \\ t:\ \text{odd}}} \sum_{I \subseteq S,\ |I|=t} LB_2(I) \ - \sum_{\substack{t_{\text{odd}}+1 \leq t \leq |S| \\ t:\ \text{even}}} \sum_{I \subseteq S,\ |I|=t} UB_2(I),$$

where we assume that $|S| \geq 3$ and $3 \leq t_{\text{odd}} \leq |S|$ is odd. Let $I^{(1)}_{\text{odd}}$ be the set containing all odd integers $3 \leq t_{\text{odd}} \leq |S|$ with $L^{(1)}(t_{\text{odd}}) < 0$. Let $t^{(1),*}_{\text{odd}}$ be the smallest integer in $I^{(1)}_{\text{odd}}$ with $L^{(1)}(t^{(1),*}_{\text{odd}}) \geq L^{(1)}(t)$ for all $t \in I^{(1)}_{\text{odd}}$. Then, by replacing I_{odd}, t^{*}_{odd} from definition 5.1.12 with $I^{(1)}_{\text{odd}}$ and $t^{(1),*}_{\text{odd}}$, we obtain a lower bound for $|M^{(1),c}|$ resp. $|M^{(-1),c}|$.

Definition 5.1.16 (Lower Bound of $|M^{(1),c}|$ resp. $|M^{(-1),c}|$)
Let $I^{(1)}_{odd}$ *and* $t^{(1),*}_{odd}$ *be as discussed above. Then, we define the lower bound of* $|M^{(1),c}|$ *resp.* $|M^{(-1),c}|$ *as follows:*

Case 1: $I^{(1)}_{odd} = \emptyset$ *or* $|S| < 3$*:*
In this case we have to consider all $I \subseteq S$ *and obtain*

$$LB_2 = \sum_{\substack{1 \leq t \leq |S| \\ t:\ odd}} \sum_{I \subseteq S,\ |I|=t} LB_2(I) \ - \sum_{\substack{2 \leq t \leq |S| \\ t:\ even}} \sum_{I \subseteq S,\ |I|=t} UB_2(I).$$

Case 2: $I^{(1)}_{odd} \neq \emptyset$ *and* $|S| \geq 3$*:*
In this case we only consider $I \subseteq S$ *with* $|I| \leq t^{(1),*}_{odd} - 1$ *and obtain*

$$LB_2 = \sum_{\substack{1 \leq t \leq t^{(1),*}_{odd}-2 \\ t:\ odd}} \sum_{I \subseteq S,\ |I|=t} LB_2(I) \ - \sum_{\substack{2 \leq t \leq t^{(1),*}_{odd}-1 \\ t:\ even}} \sum_{I \subseteq S,\ |I|=t} UB_2(I).$$

From the construction of LB_2 and UB_2, we obtain the following proposition.

Proposition 5.1.17 (Bounds of $|M^{(1),c}|$ resp. $|M^{(-1),c}|$)
Let LB_2 *and* UB_2 *be as defined above. Then we have*

$$LB_2 \leq |M^{(-1),c}| \leq UB_2, \quad \textit{resp. } LB_2 \leq |M^{(1),c}| \leq UB_2.$$

Furthermore, since $|M^{(1),c}| = |M^{(-1),c}|$ and $|A^{(1)}| = |A^{(-1)}|$, we have that

$$\begin{aligned} & 2LB_2 \leq |M^{(1),c}| + |M^{(-1),c}| \leq 2UB_2, \\ \iff & 2LB_2 \leq 2(|A^{(1)}| - |M^{(1)}|) \leq 2UB_2, \\ \iff & -2|A^{(1)}| + 2LB_2 \leq -|M^{(1)}| - |M^{(-1)}| \leq -2|A^{(1)}| + 2UB_2. \end{aligned} \tag{5.18}$$

Recall that by Proposition 5.1.2, we can rewrite $\sum_{w\in W}\mu(w)$ as

$$\sum_{w\in W}\mu(w) = (q+1)\varphi(q^4-1) - |\mathcal{M}| - |M^{(1)}| - |M^{(-1)}|.$$

Recall also that we know the cardinalities $|\mathcal{A}|$, $|A^{(1)}|$ and $|A^{(-1)}|$. Those are namely

$$|\mathcal{A}| = \frac{(q-3)(q^4-1)}{2},$$
$$|A^{(1)}| = |A^{(-1)}| = \frac{q(q-1)(q^2+1)}{2}.$$

Hence, by combining these facts with the inequalities in (5.17) and (5.18) and by also including the trivial bounds from Lemma 5.1.1, we obtain the following bounds on $\sum_{w\in W}\mu(w)$.

Theorem 5.1.18
Let LB_1, LB_2 and UB_1, UB_2 be as discussed above. We define

$$LB = \max\Big\{(q+1)\varphi(q^4-1) - \frac{(q^4-1)(q-3)}{2} - q(q-1)(q^2+1) \\ +(q-3)LB_1 + 2LB_2,\ \ 2\varphi(q^4-1)\Big\},$$

and

$$UB = \min\Big\{(q+1)\varphi(q^4-1) - \frac{(q^4-1)(q-3)}{2} - q(q-1)(q^2+1) \\ +(q-3)UB_1 + 2UB_2,\ \ (q+1)\varphi(q^4-1)\Big\},$$

Then we have

$$LB \le \sum_{w\in W}\mu(w) \le UB.$$

Recall that $2\varphi(q^4-1)$ and $(q+1)\varphi(q^4-1)$ are the trivial lower and upper bound of $\sum_{w\in W}\mu(w)$ according to Lemma 5.1.1. In case the calculations of LB_1, LB_2, UB_1, and UB_2 yield a worse bound than the trivial bound, we set accordingly the bound LB or UB as the trivial bound.
We have found some bounds on $\sum_{w\in W}\mu(w)$. The next step is to test these bounds. Since we know that there is no sophisticated way of calculating $\mu(w)$ for $w\in W$, we have to consider small values of q and calculate $\sum_{w\in W}\mu(w)$ by generating $\mathbb{F}_{q^4}^*$ and by checking each element of $\mathcal{P}(w) = \{1+a\eta^{q-1} \ : \ a\in\mathbb{F}_q\}\cup\{\eta^{q-1}\}$ if it is in the set of squares of $\mathbb{F}_{q^4}^*$ for all $w\in W$.

5.2 Performance of LB and UB for Small q.

In this section, we want to test the lower and upper bounds from Theorem 5.1.18. To do so, we want to split the bounds so that we don't take the maximum resp. the minimum and consider all cases. We will compare the following integers for small $q \equiv 3 \mod 4$ up to $q = 43$:

$$
\begin{aligned}
TLB &= 2\varphi(q^4-1),\\
TUB &= (q+1)\varphi(q^4-1),\\
LB &= (q+1)\varphi(q^4-1) - \frac{(q^4-1)(q-3)}{2} - q(q-1)(q^2+1) + (q-3)LB_1 + 2LB_2,\\
UB &= (q+1)\varphi(q^4-1) - \frac{(q^4-1)(q-3)}{2} - q(q-1)(q^2+1) + (q-3)UB_1 + UB_2,\\
S(\mu) &= \sum_{w\in W} \mu(w).
\end{aligned}
$$

Notice that using the above notation, Theorem 5.1.18 becomes

$$\max(TLB, LB) \leq S(\mu) \leq \min(TUB, UB).$$

Using the Matlab code from Appendix A.1 and Appendix A.2, we obtain the following table:

$\mathbf{q \equiv 3 \mod 4}$	**TLB**	**TUB**	**LB**	**S(μ)**	**UB**
3	64	128	68	96	116
7	1280	5120	−1780	3264	8142
11	7680	46 080	−25 900	26 880	69 658
19	69 120	691 200	−475 164	380 160	921 918
23	133 120	1 597 440	−1 469 140	867 200	2 613 882
31	442 368	7 077 888	−5 809 060	3 757 568	11 098 202
43	1 382 400	30 412 800	−41 304 300	15 869 952	61 353 130

Table 5.1: In this table, we see all the values of the trivial lower bound (TLB), the trivial upper bound (TUB), the lower bound (LB), the sum of the μ-invariants ($S(\mu)$) and the upper bound (UB) for all primes $q \leq 43$ with the property that $q \equiv 3 \mod 4$.

The key observation in table 5.1 is that the lower bound LB is terrible. Moreover, we see that the trivial bound TLB outperforms the lower bound LB. On the other hand, the upper bound works but is not significantly better than the trivial upper bound TUB. Furthermore, we see that the TUB is better than UB in all cases where $q > 3$. Figure 5.2 helps to illustrate this fact where we normalize each value with respect to $S(\mu)$, meaning that we multiply all values by $1/S(\mu)$.
We conclude that the trivial bounds are much more useful than the bounds we established in Theorem 5.1.18. Recall that we need no assumptions for the trivial upper bound since it only depends on the cardinality of $\mathcal{P}(w)$. Regarding the trivial lower bound, we needed only the assumption that the N satisfying $N \cdot n = q^r - 1$ is even but even if that was not the case, the trivial bound would change to $1 \cdot \varphi(q^r - 1)$ since $1 \in \mathcal{P}(w)$ is always a square for all primitive elements $w \in W$, which would still outperform the lower bound LB. The following example illustrates why the lower bound LB fails.

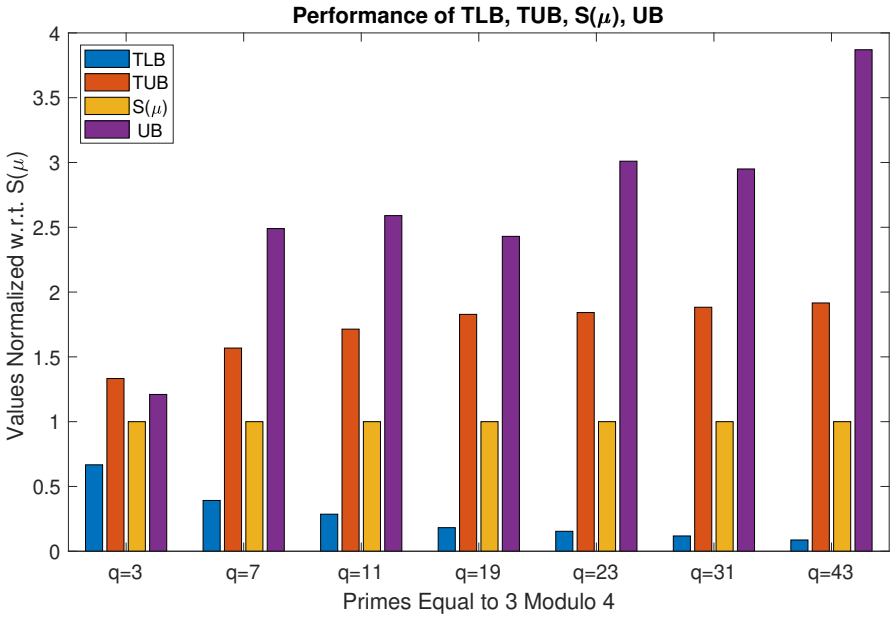

Figure 5.2: This plot visualizes table 5.1 by multiplying all values with $1/S(\mu)$. The bars represent the different bounds TLB, TUB, UB and $S(\mu)$ itself. The lower bound LB is not represented since it is always negative except for $q = 3$. We observe that the trivial bounds outperform the bounds LB and UB.

Example 5.2.1 (Calculating LB for $q = 7$)
We consider LB for the field $\mathbb{F}_{7^4}$. The factors of $7^4 - 1 = 2400$ correspond to the multiset $S = \{2, 2, 2, 2, 2, 3, 5, 5\}$. We calculate $LB_1(I)$ for $I \subseteq S$ with $|I| = t$ for $t \geq 1$.

$t = 1$: In this case, we have $I_{1,1} = \{2\}$, $I_{1,2} = \{3\}$, $I_{1,3} = \{5\}$. This gives us

$$LB_1(I_{1,1}) = \max\left(\left\lceil \frac{7^4 + 2 \cdot 7^2 - 3}{2 \cdot 2} \right\rceil - \frac{7-1}{2}7^2, 0\right) = \max(477, 0) = 477,$$
$$LB_1(I_{1,2}) = \max\left(\frac{7^4 - 1}{2 \cdot 3} - \frac{7-1}{2}7^2, 0\right) = \max(253, 0) = 253,$$
$$LB_1(I_{1,3}) = \max\left(\frac{7^4 - 1}{2 \cdot 5} - \frac{7-1}{2}7^2, 0\right) = \max(93, 0) = 93.$$

$t = 2$: In this case, we have

$$I_{2,1} = \{2, 2\},\ I_{2,2} = \{2, 3\},\ I_{2,3} = \{2, 5\},\ I_{2,4} = \{3, 5\},\ I_{2,5} = \{5, 5\}.$$

This gives us

$$LB_1(I_{2,1}) = \max\left(\left\lceil \frac{7^4+2\cdot 7^2-3}{2\cdot 4}\right\rceil - \frac{7-1}{2}7^2, 0\right) = \max(165,0) = 165,$$
$$LB_1(I_{2,2}) = \max\left(\left\lceil \frac{7^4+2\cdot 7^2-3}{2\cdot 6}\right\rceil - \frac{7-1}{2}7^2, 0\right) = \max(61,0) = 61,$$
$$LB_1(I_{2,3}) = \max\left(\left\lceil \frac{7^4+2\cdot 7^2-3}{2\cdot 10}\right\rceil - \frac{7-1}{2}7^2, 0\right) = \max(-22,0) = 0,$$
$$LB_1(I_{2,4}) = \max\left(\frac{7^4-1}{2\cdot 15} - \frac{7-1}{2}7^2, 0\right) = \max(-67,0) = 0,$$
$$LB_1(I_{2,5}) = \max\left(\frac{7^4-1}{2\cdot 25} - \frac{7-1}{2}7^2, 0\right) = \max(-99,0) = 0.$$

$t = 3$*: In this case we have*

$$I_{3,1} = \{2,2,2\},\ I_{3,2} = \{2,2,3\},\ I_{3,3} = \{2,2,5\},$$
$$I_{3,4} = \{2,3,5\},\ I_{3,5} = \{2,5,5\},\ I_{3,6} = \{3,5,5\}.$$

This gives us

$$LB_1(I_{3,1}) = \max\left(\left\lceil \frac{7^4+2\cdot 7^2-3}{2\cdot 8}\right\rceil - \frac{7-1}{2}7^2, 0\right) = \max(9,0) = 9,$$
$$LB_1(I_{3,2}) = LB_1(I_{3,3}) = LB_1(I_{3,4}) = LB_1(I_{3,5}) = LB_1(I_{3,6}) = 0.$$

This also implies that $LB_1(I) = 0$ for all $I \subseteq S$ with $|I| \geq 4$. Moreover, since the $UB_1(I)$ with $|I| = 4$ are much larger than the $UB_1(I)$ with $|I| = 6, 8$, we have that $I_{odd} = \{3,5,7\}$ with $t^*_{odd} = 3$. Therefore, we need $UB_1(I)$ for all $I \subseteq S$ with $|I| = 2$.

$$UB_1(I_{2,1}) = \min\left(\left\lfloor \frac{7^4-2\cdot 7^2-3}{2\cdot 4}\right\rfloor + \frac{7-1}{2}7^2, \frac{7^4-1}{4}\right) = \min(434,600) = 434,$$
$$UB_1(I_{2,2}) = \min\left(\left\lfloor \frac{7^4-2\cdot 7^2-3}{2\cdot 6}\right\rfloor + \frac{7-1}{2}7^2, \frac{7^4-1}{6}\right) = \min(338,400) = 338,$$
$$UB_1(I_{2,3}) = \min\left(\left\lfloor \frac{7^4-2\cdot 7^2-3}{2\cdot 10}\right\rfloor + \frac{7-1}{2}7^2, \frac{7^4-1}{10}\right) = \min(262,240) = 240,$$
$$UB_1(I_{2,4}) = \min\left(\frac{7^4-1}{2\cdot 15} + \frac{7-1}{2}7^2, \frac{7^4-1}{15}\right) = \min(227,160) = 160.$$
$$UB_1(I_{2,5}) = \min\left(\frac{7^4-1}{2\cdot 25} + \frac{7-1}{2}7^2, \frac{7^4-1}{25}\right) = \min(195,96) = 96.$$

This gives us

$$LB_1 = \max\left(\sum_{I\subseteq S\,:\,|I|=1} LB_1(I) - \sum_{I\subseteq S\,:\,|I|=2} UB_1(I), 0\right) = \max(-445,0) = 0.$$

Analog we can compute LB_2 by using $LB_2(I)$ and $UB_2(I)$ from Proposition 5.1.14 and we obtain

$$LB_2 = \max(-448, 0) = 0.$$

We conclude that

$$LB = (7+1)\varphi(7^4-1) - \frac{(7^4-1)(7-3)}{2} - 7(7-1)(7^2+1) + 0 + 0 = -1780.$$

The $UB_1(I)$ and $UB_2(I)$ terms dominate the terms of $LB_1(I)$ and $LB_2(I)$ such that all cases where $q > 3$ in table 5.1, LB_1 and LB_2 equal zero. The dominance of $UB_1(I)$ and $UB_2(I)$ follows from the fact that, in general, there are more subsets $I \subseteq S$ with $|I| = 2$ than with $|I| = 1$ since there are more possibilities to choose 2 factors from S than to choose only one.

Considering the figure 5.2, we observe that $S(\mu)$ is always between the trivial bounds. This leads to the following estimate for $S(\mu)$.

Proposition 5.2.2 (Estimate of $S(\mu)$)
Let $q \leq 43$ be a prime such that $q \equiv 3 \mod 4$. We define the estimation of $S(\mu)$ as

$$\tilde{S}(\mu) = \frac{TLB + TUB}{2} = \frac{(q+3)\varphi(q^4-1)}{2}. \tag{5.19}$$

Then we have that

$$\left| 1 - \frac{\tilde{S}(\mu)}{S(\mu)} \right| \leq 0.02.$$

The plot 5.3 illustrates Proposition 5.2.2. It remains an open question if the estimate works for all $q \geq 3$ with $q \equiv 3 \mod 4$.
We conclude that the bounds from Theorem 5.1.18 work but are most likely just the trivial bounds. The advantage of the trivial bounds is that they work for a larger class of b-symbol codes. We generalize this fact for b-symbol codes in the following theorem. Recall that the cardinality of $\mathcal{P}(w)$ for $b \geq 2$ is given by $|\mathcal{P}(w)| = \frac{q^b-1}{q-1}$.

Theorem 5.2.3 (Trivial Bounds for $u = 2$)
Let $N = \frac{q^r-1}{n}$, where n is the length of the code $\mathcal{C}$ and $r \geq 2$ is an even integer and the size of the code $\mathcal{C}$. Assume that $u = \gcd((q^r-1)/(q-1), N) = 2$ holds. Let $w \in W$ be a primitive element in $\mathbb{F}_{q^r}^*$, then the sum of μ-invariant satisfies:

***Case 1:** N is even:*

$$b \cdot \varphi(q^r-1) \leq \sum_{w \in W} \mu(w) \leq \frac{(q^b-1) \cdot \varphi(q^r-1)}{(q-1)}.$$

***Case 2:** N is odd:*

$$\left\lceil \frac{b}{2} \right\rceil \cdot \varphi(q^r-1) \leq \sum_{w \in W} \mu(w) \leq \frac{(q^b-1) \cdot \varphi(q^r-1)}{(q-1)}.$$

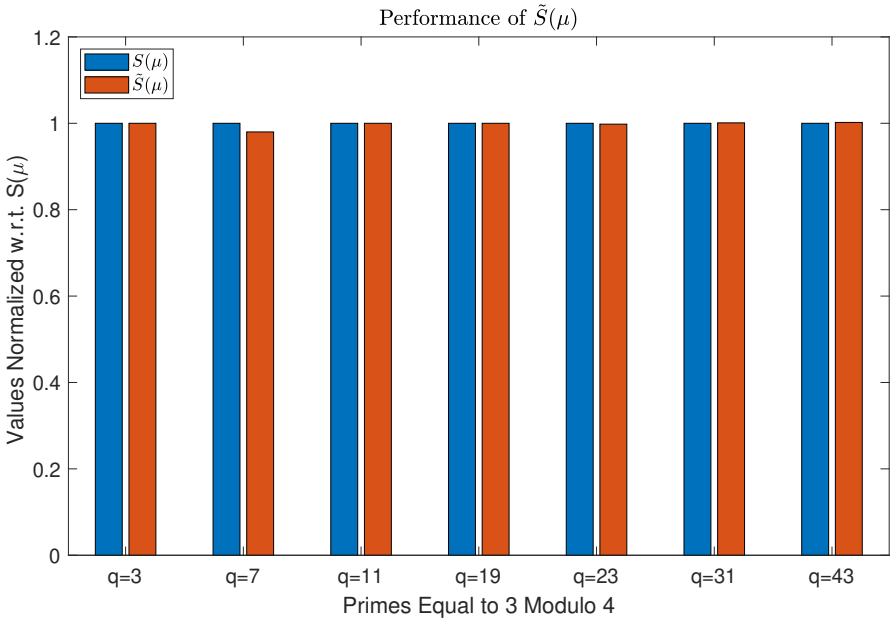

Figure 5.3: This plot visualizes Proposition 5.2.2 by multiplying $S(\mu)$ and $\tilde{S}(\mu)$ with $1/S(\mu)$. We observe that the estimate $\tilde{S}(\mu)$ is a good approximation in these cases.

Proof. The upper bound follows immediately by multiplying the cardinality of $\mathcal{P}(w)$, which is the maximal possible value for $\mu(w)$, with the number of primitive elements in $\mathbb{F}_{q^r}^*$. Recall that $\mathcal{P}(w)$ is defined through

$$\mathcal{P}(w) = \bigcup_{j=1}^{b-1} \{w^{(j-1)N} + x_1 w^{jN} + \cdots + x_{b-j} w^{(b-j)N} \ : \ x_1, ..., x_{b-j} \in \mathbb{F}_q\} \cup \{w^{(b-1)N}\}.$$

This implies that $w^0, w^N, w^{2N}, ..., w^{(b-1)N} \in \mathcal{P}(w)$. Now, if N is even, then clearly, for all $w \in W$, all elements $w^0, ..., w^{(b-1)N}$ are squares in $\mathcal{P}(w)$, which gives us the lower bound in the case 1 scenario. If N is odd, then we have two possible scenarios for the elements $w^0, ..., w^{(b-1)N}$. Assume b is even then for all $w \in W$ the elements $w^0, w^{2N}, w^{4N} ..., w^{(b-2)N}$, which are in total $\frac{b}{2}$ elements, are squares in $\mathcal{P}(w)$. Assume now that b is odd, then the elements $w^0, w^{2N}, w^{4N} ..., w^{(b-1)N}$, which are in total $\lceil \frac{b}{2} \rceil$ elements, are squares in $\mathcal{P}(w)$. Combining both scenarios, we obtain the lower bound of case 2. □

Theorem 5.2.3 is a generalization of Lemma 5.1.1 and gives us some lower and upper bounds for all b-symbol codes over semiprimitive irreducible cyclic codes where $\gcd((q^r - 1)/(q-1), N) = 2$. Similarly, one can deduce trivial bounds for all semiprimitive irreducible cyclic codes. Recall that for $0 \leq i \leq u-1$, we have $C_i^{(u,q^r)} = w^i \langle w^u \rangle$ and

$$\mu_{(i)}(b) = |\{x \in \mathcal{P}(b) \ : \ x \in C_i^{(u,q^r)}\}|.$$

Theorem 5.2.4 (Trivial Bounds for all u)
Let $\mathcal{C}$ be an (n, r) semiprimitive irreducible cyclic code with $N = (q^r - 1)/n$, $r \geq 2$ even and $u = \gcd((q^r - 1)/(q - 1), N)$ such that $u \leq b$. Let $w \in W$ be a primitive element in $\mathbb{F}_{q^r}^*$. Let $b \equiv t \mod u$. Then, the sum of μ-invariants satisfies:

Case 1: $N \equiv i \mod u$*:*

$$b \cdot \varphi(q^r - 1) \leq \sum_{w \in W} \mu_{(i)}(w) \leq \frac{(q^b - 1) \cdot \varphi(q^r - 1)}{(q - 1)}.$$

Case 2: $N \not\equiv i \mod u$, $i < t$*:*

$$\left\lceil \frac{b}{u} \right\rceil \cdot \varphi(q^r - 1) \leq \sum_{w \in W} \mu_{(i)}(w) \leq \frac{(q^b - 1) \cdot \varphi(q^r - 1)}{(q - 1)}.$$

Case 3: $N \not\equiv i \mod u$, $i \geq t$*:*

$$\left\lfloor \frac{b}{u} \right\rfloor \cdot \varphi(q^r - 1) \leq \sum_{w \in W} \mu_{(i)}(w) \leq \frac{(q^b - 1) \cdot \varphi(q^r - 1)}{(q - 1)}.$$

Proof. The proof is very similar to the proof of Theorem 5.2.3. The argument for the upper bound is the same. For the lower bound, we consider again the elements $w^0, w^N, ..., w^{(b-1)N} \in \mathcal{P}(b)$. Note that for $0 \leq j \leq b - 1$

$$w^{jN} \in C_i^{(u,q^r)} \iff jN \equiv i \mod u.$$

Therefore, if $N \equiv i \mod u$, we have the lower bound of the case 1 scenario. If $N \not\equiv i \mod u$, then we must count the cases where $j \equiv i \mod u$. Using the fact that $u \leq b$, there are either $\lfloor \frac{b}{u} \rfloor$ or $\lceil \frac{b}{u} \rceil$ elements $j \in \{0, 1, ..., b - 1\}$ satisfying $j \equiv i \mod u$ depending on the value of t. If $i < t$, then there is an element $j \in \{b - t, b - t + 1, ..., b - 1\}$ satisfying $j \equiv i \mod u$. This implies the lower bound of the case 2 scenario. On the other hand if $i \geq t$, then there is no $j \in \{b - t, b - t + 1, ..., b - 1\}$ □

Theorem 5.2.4 gives us lower and upper bounds for all sums of μ-invariants of all semiprimitive irreducible cyclic codes where we consider the sums of the invariants $\mu_{(i)}(w)$ over all $w \in W$. In table 5.1 and in figure 5.2, we considered these bounds for $b = 2$ and $u = 2$. We observed that there is still room for improvement. One idea to improve these bounds is by investigating divisors of $\sum_{w \in W} \mu(w)$. Therefore, we will again focus on symbol-pair codes ($b = 2$) with $u = \gcd((q^r - 1)/(q - 1), N) = 2$ and $r = 4$. We will see more about that in the following section.

5.3 Divisors of the Sum of μ-Invariants

In this section, we keep the assumptions from Chapter 5. This means that we consider again symbol-pair codes over fields of the form $\mathbb{F}_{q^4}$ with $q \equiv 3 \mod 4$. The goal is to provide some divisors of $S(\mu) = \sum_{w \in W} \mu(w)$ which can be used to improve its lower and upper bounds from Theorem 5.1.18. The following lemma implies that $S(\mu)$ is even.

Lemma 5.3.1 ($\mu(w) = \mu(w^{-1})$)
Let $w \in W$, then we have also that $w^{-1} \in W$ and

$$\mu(w) = \mu(w^{-1}).$$

Proof. Clearly, if $w = \eta^i$ with $\gcd(i, q^4 - 1) = 1$, then we also have $\gcd(-i, q^4 - 1) = 1$, which implies that $w^{-1} \in W$. Now, let $\psi : P(w) \to P(w^{-1})$ be defined through

$$w^{q-1} \mapsto w^{-(q-1)}, \text{ and } 1 + aw^{q-1} \mapsto 1 + a^{-1}w^{-(q-1)},$$

where $a \in \mathbb{F}_q$. Then by construction ψ is a bijection between $P(w)$ and $P(w^{-1})$. Suppose now $1 + aw^{q-1} = y^2$ for some $a \in \mathbb{F}_q$ and $y \in \mathbb{F}_{q^4}^*$. We have

$$\begin{aligned} 1 + aw^{q-1} &= y^2 && \mid \cdot\, w^{-(q-1)} \\ w^{-(q-1)} + aw^{q-1}w^{-(q-1)} &= y^2 w^{-(q-1)} && \\ a + w^{-(q-1)} &= y^2 w^{-(q-1)} && \mid \cdot\, a^{-1} \\ 1 + a^{-1}w^{-(q-1)} &= y^2 w^{-(q-1)} a^{-1}. && \end{aligned}$$

Since on the right-hand side, we clearly have a square, we conclude that ψ sends squares from $P(w)$ to squares in $P(w^{-1})$. On the other hand, if we assume that $1 + aw^{q-1} = w^{2k+1}$ for some integer k. Then, by the same argument as above, we have that $1 + a^{-1}w^{-(q-1)}$ is not a square in $\mathbb{F}_{q^4}^*$. Hence, ψ maps non-squares to non-squares. Since ψ is a bijection, we conclude $\mu(w) = \mu(w^{-1})$. □

The next lemma expands the idea of Lemma 5.3.1 for a larger set of primitive elements generating the same set $\mathcal{P}(w)$.

Lemma 5.3.2
There exists an even integer M such that for all positive integer i coprime to $q^4 - 1$, we have

$$\left| \left\{ 0 \le k \le q - 2 \ : \ \gcd\left(i + k\frac{q^4 - 1}{q - 1}, q^4 - 1 \right) = 1 \right\} \right| = M.$$

This means that we have M primitive elements of the form

$$w_1 = \eta^{i + k_1 \frac{q^4-1}{q-1}}, ..., w_M = \eta^{i + k_M \frac{q^4-1}{q-1}},$$

for M distinct $k_1, ..., k_M \in \{0, 1, ..., q - 3, q - 2\}$ such that

$$\mathcal{P}(w_1) = \mathcal{P}(w_2) = \cdots = \mathcal{P}(w_M),$$

and

$$\mu(w_1) = \mu(w_2) = \cdots = \mu(w_M).$$

Proof. Let w^i be a primitive element in $\mathbb{F}_{q^4}^*$. Then for $0 \le k \le q - 2$ we have that

$$\left(w^{i + k\frac{q^4-1}{q-1}}\right)^{q-1} = \left(w^i\right)^{q-1}.$$

This implies that if $\gcd(i + k\frac{q^4-1}{q-1}, q^4 - 1) = 1$, then we have that

$$\mathcal{P}(w^i) = \mathcal{P}(w^{i+k\frac{q^4-1}{q-1}}) \text{ and } \mu(w^i) = \mu(w^{i+k\frac{q^4-1}{q-1}}).$$

Let $\omega = \frac{q^4-1}{q-1}$. The question is now for which k we have that $i + k\omega$ and $q^4 - 1$ are coprime. Note that

$$\begin{aligned} & \gcd(i + k\omega, q^4 - 1) = 1, \\ \iff & \gcd(i + k\omega, \omega(q - 1)) = 1, \\ \iff & \gcd(i + k\omega, \omega) = 1, \text{ and } \gcd(i + k\omega, q - 1) = 1, \\ \iff & \gcd(i, \omega) = 1, \text{ and } \gcd(i + k\omega \mod q - 1, q - 1) = 1. \end{aligned}$$

Since $\gcd(i, q^4 - 1) = 1$, we have that $\gcd(i, \omega) = 1$. It remains to show that

$$|\{0 \leq k \leq q - 2 \ : \ \gcd(i + k\omega \mod q - 1, q - 1) = 1\}| = M \tag{5.20}$$

for all i coprime to $q^4 - 1$. Using the fact that

$$\omega = \frac{q^4 - 1}{q - 1} = q^3 + q^2 + q + 1 \equiv 4 \mod q - 1,$$

the equation (5.20) is equivalent to

$$|\{0 \leq k \leq q - 2 \ : \ \gcd(i + 4k \mod q - 1, q - 1) = 1\}| = M. \tag{5.21}$$

We split equation (5.21) in two cases and show that in both cases, the equation (5.21) holds.

Case 1: $gcd(4, q - 1) = 2$:
In this case, we have that $4x \equiv 2 \mod q - 1$ for some integer x and therefore the sequence $\{4k \mod q - 1 \ : \ 0 \leq k \leq q - 2\}$ goes twice through all even numbers up to $q - 1$. I.e., we obtain the sequence

$$\{4k \mod q-1\}_{0\leq k\leq q-2} = \{0, 4, 8, ..., q-3, 2, 6, 10, ..., q-5, 0, 4, 8, ..., q-3, 2, 6, 10, ..., q-5\}.$$

Thus, we conclude that for all integers i with $gcd(i, q^4 - 1) = 1$ the sequence $\{i + 4k \mod q - 1\}_{0\leq k\leq q-2}$ goes twice through all odd numbers less or equal to $q - 2$. Therefore, there is an even number M such that for each i coprime to $q^4 - 1$, M is the solution to the equation (5.21).

Case 2: $gcd(4, q - 1) = 4$:
Analog to the case above, we can consider the sequence $\{4k \mod q - 1\}_{0\leq k\leq q-2}$.

$$\{4k \mod q - 1\}_{0\leq k\leq q-2} = \{0, 4, ..., q - 5, 0, 4, ..., q - 5, 0, 4, ..., q - 5, 0, 4, ..., q - 5\}$$

Suppose $i \equiv 1 \mod 4$, then $\{i + 4k \mod q - 1\}_{0\leq k\leq q-2}$ generates the sequence

$$\{1, 5, 9, ..., q - 4, 1, 5, 9, ..., q - 4, 1, 5, 9, ..., q - 4, 1, 5, 9, ..., q - 4\},$$

and for each such i we have the same solution to equation (5.21). Suppose now $i \equiv 3 \mod 4$, then the sequence $\{i + 4k \mod q-1\}_{0\leq k\leq q-2}$ consists of all the additive inverses of the $i \equiv 1 \mod 4$ case in $\mathbb{Z}_{q-1}$. I.e. $\{i + 4k \mod q-1\}_{0\leq k\leq q-2}$ generates

$$\{3, 7, 11, ..., q-2, 3, 7, 11, ..., q-2, 3, 7, 11, ..., q-2, 3, 7, 11, ..., q-2\}.$$

Using the fact that for two positive integers a, b, we have $gcd(a, b) = gcd(b - a, b)$, we conclude that the $i \equiv 3 \mod 4$ case has the same amount of solutions as the $i \equiv 1 \mod 4$ case. Therefore, there exists an even number M such that for all i coprime to $q^4 - 1$, M is the solution to equation (5.21).

□

Notice that we only need the sequence $\{1+4k \mod q-1\}_{0\leq k\leq q-2}$ to determine M. We see that in the following example.

Example 5.3.3
Let $q = 19$, then we obtain the sequence

$$\{1 + 4k \mod 18\}_{0\leq k\leq 17} = \{1, 5, 9, 13, 17, 3, 7, 11, 15, 1, 5, 9, 13, 17, 3, 7, 11, 15\}$$

considering only the numbers which are coprime to 18, we have

$$M = 2 \cdot |\{1, 5, 13, 17, 7, 11, \}| = 12.$$

Let $q = 29$, then $\{1 + 4k \mod 28\}_{0\leq k\leq 27}$ gives us the sequence

$$\{1, 5, 9, 13, 17, 21, 25, 1, 5, 9, 13, 17, 21, 25, 1, 5, 9, 13, 17, 21, 25, 1, 5, 9, 13, 17, 21, 25\}$$

We can calculate M as

$$M = 4 \cdot |\{1, 5, 9, 13, 17, 25\}| = 24$$

This example gives us an idea of how one could directly compute M using the Euler phi function.

Corollary 5.3.4 (Calculate M Directly)
Let M be the solution of equation (5.21), then one can calculate M directly using the Euler phi function by

$$M = 2 \cdot \varphi(q - 1). \tag{5.22}$$

Proof. The proof follows directly from the proof of Lemma 5.3.2. Notice that in the case where $\gcd(4, q-1) = 2$ the sequence $\{1+4k \mod q-1\}_{0\leq k\leq q-2}$ repeats itself twice through all odd numbers up to $q - 2$. Since $q - 1$ is even, all numbers coprime to $q - 1$ must be odd. Therefore, the number of elements in $\{1 + 4k \mod q-1\}_{0\leq k\leq q-2}$ which are coprime to $q - 1$ are $2\varphi(q - 1)$.
On the other hand, if we have $\gcd(4, q - 1) = 4$. Then the sequence $\{1 + 4k \mod q - 1\}_{0\leq k\leq q-2}$ goes 4 times through one-half of the odd numbers up to $q - 1$. Since the other half of odd numbers which are not in $\{1 + 4k \mod q-1\}_{0\leq k\leq q-2}$ are their additive inverse in $\mathbb{Z}_{q-1}$, and $q - 1$ is even, we must have that exactly one-half of the numbers coprime to $q - 1$ are in $\{1 + 4k \mod q-1\}_{0\leq k\leq q-2}$. This gives us

$$M = 4 \cdot \frac{\varphi(q - 1)}{2} = 2 \cdot \varphi(q - 1).$$

□

The following theorem combines Lemma 5.3.2 and Corollary 5.3.4 and states that $2\varphi(q-1)$ divides $S(\mu)$.

Theorem 5.3.5
Let M be the solution to equation (5.21). Then $M = 2\varphi(q-1)$ and

$$M \Big| \sum_{w \in W} \mu(w)$$

Proof. The proof follows directly from Lemma 5.3.2 and Corollary 5.3.4. □

Example 5.3.6
Consider the field $\mathbb{F}_{19^4}$. We have $\sum_{w \in W} \mu(w) = 380160$ and $M = 2\varphi(18) = 12$ which gives us

$$\frac{\sum_{w \in W} \mu(w)}{M} = \frac{380160}{12} = 31680.$$

Using Theorem 5.3.5, we can improve any bounds on $\sum_{w \in W} \mu(w)$ in the following way.

Corollary 5.3.7 (Improvement of Bounds)
Let LB and UB be positive integers such that

$$LB \leq \sum_{w \in W} \mu(w) \leq UB.$$

Let $M = 2\varphi(q-1)$. Then, we can define

$$\mathcal{LB} = \left\lceil \frac{LB}{M} \right\rceil \cdot M, \quad \text{and} \quad \mathcal{UB} = \left\lfloor \frac{UB}{M} \right\rfloor \cdot M,$$

and obtain

$$LB \leq \mathcal{LB} \leq \sum_{w \in W} \mu(w) \leq \mathcal{UB} \leq UB.$$

We conclude that we have found some useful divisors of $S(\mu)$. These divisors allow us to improve any lower and upper bounds according to Corollary 5.3.7. Note that $M = 2\varphi(q-1)$ is not large enough to significantly improve the bounds from Theorem 5.1.18. Therefore, there is still room for improvement regarding the bounds. Nevertheless, by looking for divisors of $\sum_{w \in W} \mu(w)$, we have found some useful properties. For instance, we know now that for all primitive elements $w \in W$, we have that $\mu(w) = \mu(w^{-1})$, where we only used the fact that $a \in \mathbb{F}_q$ is a square in $\mathbb{F}_{q^r}$ for even $r \geq 2$ and that $N = q-1$ is even. It is important to emphasize that our results are based on a simplified setting. Therefore, it remains to show if one can generalize Theorem 5.3.5 for a larger class of b-symbol codes.
Recall that our goal in this Chapter was to investigate the invariant $\mu(b)$, which is the last missing puzzle in finding the b-symbol weight distribution of many irreducible cyclic codes. We considered the simplest setting where this invariant is not trivial, where trivial means that $b = r$. Therefore, we considered symbol-pair codes ($b = 2$) over the field $\mathbb{F}_{q^4}$. Since $\mu(b)$ depends on the choice of the primitive element, we investigated the sum of all μ-invariants and deduced bounds, an estimate, and divisors of it. The goal of the next Chapter is to summarize all the results and to conclude this thesis.

Chapter 6

Conclusion

Our main goal was to determine the b-symbol weight distribution of semiprimitive irreducible cyclic codes, which are a large class within the irreducible cyclic codes. The b-symbol weight distribution is determined up to an invariant $\mu(b)$ resp. $\mu_{(i)}(b)$ as we have seen in Theorem 4.4.8 and 4.4.14. It is still an open problem to determine these invariants or to have good lower and upper bounds. Therefore, our main goal consists of investigating these invariants. We know that for any (n, r) semiprimitive irreducible cyclic code $\mathcal{C} \subseteq (\mathbb{F}_q)^n$, $\mu(r)$ resp. $\mu_{(i)}(r)$ is determined according to Lemma 4.4.4 resp. Lemma 4.4.17 and considered as the trivial case for the μ-invariant. Therefore, we studied the simplest cases where these μ-invariants are not trivial, which is the case when $b = 2$ and $r = 4$. While investigating this issue, we encountered the problem that the μ-invariants depend on the choice of the primitive element, which is used to describe the code $\mathcal{C}$. Therefore, we considered the average value of the μ-invariants by investigating the sum of the μ-invariants over all primitive elements of $\mathbb{F}_{q^4}$.

We have found many bounds of the sum of μ-invariants as we have seen in Lemma 5.1.1 and Theorem 5.1.18. We have tested those bounds for small primes up to $q = 43$ and observed that while these bounds work, there is still plenty of room for improvement. In Theorem 5.2.4, we also achieved to generalize the bounds from Lemma 5.1.1 for all semiprimitive irreducible cyclic codes. In Proposition 5.2.2, we have found a very good estimate of the sum of μ-invariants with a relative error of less than 2% for all $q \leq 43$ with $q \equiv 3 \mod 4$. In Theorem 5.3.5, we established a useful divisor of the sum of μ-invariants. We know now that $2\varphi(q-1)$ divides the sum of μ-invariants, which can be used to improve any bound according to Corollary 5.3.7.

The bounds and the estimate of the sum of μ-invariants give us information about the average value of $\mu(b)$ resp. $\mu_{(i)}(b)$. This, on the other hand, provides us information about the b-symbol weight distribution of semiprimitive irreducible codes.
The divisors of the sum of μ-invariants are useful to improve the lower and upper bounds according to Corollary 5.3.7. It will also be useful for a deeper number theoretic investigation on this topic, where divisors play an essential role. In all, this thesis led to a deeper understanding of the invariants $\mu(b)$ resp. $\mu_{(i)}(b)$ which is crucial for the b-symbol weight distribution of semiprimitive irreducible cyclic codes.

In this thesis, we focused on semiprimitive irreducible cyclic codes, which are a large class of irreducible cyclic codes, but there are also some irreducible cyclic codes that are not semiprimitive, and therefore, our results are not applicable to them. In order to understand the invariants $\mu(b)$ resp. $\mu_{(i)}(b)$ better, we strongly simplified the setting where we fixed $b = 2$ and $r = 4$. In other words, we studied symbol-pair codes over $(n, 4)$ irreducible cyclic codes. There are still many open questions regarding the cases where $r = 6,\ r = 8,\ r = 10, \cdots$ and $b < r$. For instance, is there some similar divisibility rule as in Theorem 5.3.5 for these cases?
To test the bounds from Theorem 5.1.18, we need to have the exact values of the sum of μ-invariants. Since there is still no formula we can use to determine the μ-invariants, we had to calculate the sum of μ-invariants by constructing for all primitive elements $w \in \mathbb{F}_{q^4}^*$ the sets $\mathcal{P}(w)$ and checking each element in $\mathcal{P}(w)$ whether it is a square or not. This is a very time-consuming process, and that is why we could test only for primes q up to 43 with $q \equiv 3 \mod 4$. Therefore, we don't know how the bounds and the estimate perform for larger primes resp. prime powers.

There are still many open questions regarding the μ-invariants. There are still no formulas to compute the exact values of $\mu(b)$ resp. $\mu_{(i)}(b)$. The tests in table 5.1 also tell us there is still the possibility to find better bounds than we did. It would also be very useful to know if the estimate in Proposition 5.2.2 works in a more general setting. For example, can we combine the trivial bounds theorem (Theorem 5.2.4) to obtain an estimate for all semiprimitive irreducible cyclic codes?

To conclude this thesis, we emphasize the importance of the invariants $\mu(b)$ resp. $\mu_{(i)}(b)$. As our technological progress increases, the importance of b-symbol codes that can handle high-density data will become essential. Therefore, knowing more about the weight distribution of these codes is crucial, especially for knowing the error-correction capacity.

Appendix A

Calculations of μ-Invariants and Bounds

This section contains the Matlab codes we used in this thesis and provides some explanation of how they work.

A.1 Calculate the Sum of μ-Invariants

The calculation of the sum of μ-invariants is straightforward. The code takes a prime value q as input and generates the field $\mathbb{F}_{q^4}$. Then, the code searches all primitive elements by looking for integers i satisfying $\gcd(i, q^4 - 1) = 1$. The code calculates for each primitive element the $q-1$-th power. According to Lemma 5.3.2, there are the same amount of duplicates for each primitive element to the $(q-1)$-th power, and we consider exactly one of them. E.g., let $w \in \mathbb{F}_{34}^*$ be a primitive element. Then w^{41} is also a primitive element, and we have

$$w^{(q-1)} = w^2 = (w^{41})^2.$$

Therefore, it is enough to consider only one primitive element w or w^{41} since $\mathcal{P}(w) = \mathcal{P}(w^{41})$. The code continues to create $\mathcal{P}(w)$ for each primitive element $w \in \mathbb{F}_{q^4}^*$ where we exclude the duplicates. Then, it checks each element in $\mathcal{P}(w)$ whether it is a square or not to calculate $\mu(w)$. In the end, we add the duplicates.

```
function [M,mu] = mu_dist(q)
    %calculate sum of all mu(w) (=M) and list of all mu(w) (=mu)
    %for all primitive elements w in F*_q^4

    %primitive polynomial of deg 4 in F_q[x]
    p = gfprimfd(4,'min',q);

    %list of all elements in F_q^4
    f_q_4 = gftuple([0:q^4-2]',p,q);

    %list of all squares in F_q^4
```

```
sq_q_4 = f_q_4(1:2:q^4-1,1:4);

coprime = [];
for i=1:q^4-1
    if gcd(i,q^4-1)==1
        coprime = [coprime i];
    end
end


phi = length(coprime);

%powers of primitive element w; w^(i*(q-1))
w_power = mod((q-1)*coprime,q^4-1);


%Use the fact that there exist primitive elements w^i and w^j
%s.t. (w^i)^(q-1) = (w^j)^(q-1), e.g. 2 = 41*2 mod 80
%e.g. 2 = 41*2 mod 80 => (w^1)^(q-1)=(w^41)^(q-1) for q=3

%This reduces the amount of elements in w_power
[duplicates, w_power] = groupcounts(w_power');

m=length(w_power);

a=zeros(m,4);

for i=1:m
    k=w_power(i);
    a(i,:)=f_q_4(k+1,:);
end

%Create P(2) for all primitive elements
P2=[];
for i=1:m
    n=a(i,:);
    for x1=0:q-1
        P2=[P2;mod([1 0 0 0]+x1*n,q)];
    end
    P2 = [P2;n];
end


duplicates = duplicates(1);
mu=zeros(1,m);
%check each element in P2 whether it is a square or not
for i=1:m
```

```
59          for j=1:q+1
60              if ismember(P2((i-1)*(q+1)+j,:),sq_q_4,"rows")
61                  mu(i) = mu(i)+1;
62              end
63          end
64      end
65      %Add all duplicates to mu (there are 2*eulerPhi(q-1))
66      mu_duplicate = mu;
67      for i=1:duplicates-1
68          mu = [mu mu_duplicate];
69      end
70
71      M=sum(mu);
```

A.2 Calculate Lower and Upper Bounds of the Sum of μ-Invariants

The following code was needed to compute LB and UB in table 5.1, where we tested the performance of the lower and upper bounds. Note that this code calculates the bounds from Theorem 5.1.18 without considering the trivial bounds. The trivial bounds, we calculate separately using the formulas

$$TLB = 2 \cdot \varphi(q^4 - 1), \quad \text{and } TUB = (q+1) \cdot \varphi(q^4 - 1).$$

For the upper bound, the code calculates UB_1 and UB_2 according to definition 5.1.11 and definition 5.1.15. For the lower bound, the code calculates LB_1 and LB_2 according to definition 5.1.12 and definition 5.1.16.

```
1  function [UpperBound] = UB_New(q)
2
3      S=factor(q^4-1);
4
5      %S=[a a a b b c ...]
6
7      n=length(S);
8
9      I=divisors(q^4-1);
10     I=I(1,2:end);
11
12     %step 1) we want to find t_star in I_even
13
14     %I_factor = |I| for all I in S
15     I_factor = zeros(1,length(I));
16     for i = 1:length(I)
17         m_ub = length(factor(I(i)));
18         I_factor(i)=m_ub;
19     end
```

```
20
21      %We want to find I_even
22      %I_even consists of integers t_even s.t. ;
23      % 2 <= t_even <= |S|, R(t_even) > 0
24
25      I_even = [];
26      R_even=[]; %R(t_even)
27      t_star = 2;
28
29      for t_even = 2:2:length(S)
30          R=[];
31
32
33          for t = t_even+1:2:length(S)
34
35              I_ub = []; %all |I| = t ; odd
36              I_lb = []; %all |I| = t-1 ; even
37
38              m_ub = []; %all m = product(I); |I| = t
39              m_lb = []; %all m = product(I); |I| = t-1
40
41              sum_ub = []; %sum over all |I| = t of UB_1(I)
42              sum_lb = []; %sum over all |I| = t-1 of LB_1(I)
43              for i = 1:length(I_factor)
44                  if I_factor(i) == t
45                      m_ub = [m_ub I(i)];
46
47                  elseif I_factor(i) == t-1
48                      m_lb = [m_lb I(i)];
49
50                  end
51              end
52
53              for j = 1:length(m_ub)
54                  if mod(m_ub(j),2) == 1
55                      m=m_ub(j);
56                      a=min((q^4-1)/(2*m)+(q-1)/2*q^2, ...
57                          (q^4-1)/m);
58                  else
59                      m=m_ub(j);
60                      a=min(floor((q^4-2*q^2-3)/(2*m)) ...
61                          +(q-1)/2*q^2,(q^4-1)/m);
62                  end
63                  sum_ub = [sum_ub a];
64              end
65
66              for j = 1:length(m_lb)
```

```
67                  if mod(m_lb(j),2) == 1
68                      m=m_lb(j);
69                      b = max((q^4-1)/(2*m)-(q-1)/2*q^2,0);
70                  else
71                      m=m_lb(j);
72                      b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
73                          -(q-1)/2*q^2,0);
74                  end
75                  sum_lb = [sum_lb b];
76              end
77
78              R_ub = sum(sum_ub);
79              R_lb = sum(sum_lb);
80
81              R = [R R_ub-R_lb];
82          end
83          d=sum(R);
84          R_even = [R_even d];
85          if d>0
86              I_even = [I_even t_even];
87          end
88      end
89
90      %Now we have R(t_even)
91      %I_even consists of t_even which are strictly positive
92
93      %now we want to find t_star
94      %t_star s.t. R(t_star) > R(t_even) for all t_even
95
96
97      R_max = max(R_even);
98      counter = 1;
99
100     while R_even(counter) ~= R_max
101         t_star = I_even(counter+1);
102         counter = counter + 1;
103     end
104
105
106     %Now we have all we need to calculate UB_1
107
108     %Assume I_even not empty
109
110     UB_1 = 0;
111
112     for t=1:2:t_star-1
113
```

```
114         m_ub = [];
115         m_lb = [];
116
117         sum_ub = [];
118         sum_lb = [];
119
120         for i=1:length(I_factor)
121             if  I_factor(i) == t
122                 m_ub = [m_ub I(i)];
123             elseif I_factor(i) == t-1
124                 m_lb = [m_lb I(i)];
125             end
126         end
127
128         for j=1:length(m_ub)
129
130             if mod(m_ub(j),2)==1
131                 m = m_ub(j);
132                 a=min((q^4-1)/(2*m)+(q-1)/2*q^2, ...
133                     (q^4-1)/m);
134             else
135                 m = m_ub(j);
136                 a=min(floor((q^4-2*q^2-3)/(2*m)) ...
137                     +(q-1)/2*q^2,(q^4-1)/m);
138             end
139             sum_ub = [sum_ub a];
140         end
141
142         for j=1:length(m_lb)
143
144             if mod(m_lb(j),2)==1
145                 m = m_lb(j);
146                 b = max((q^4-1)/(2*m)-(q-1)/2*q^2,0);
147             else
148                 m = m_lb(j);
149                 b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
150                     -(q-1)/2*q^2,0);
151             end
152             sum_lb = [sum_lb b];
153         end
154
155         UB_1 = UB_1 + sum(sum_ub)-sum(sum_lb);
156
157     end
158
159     %Analoge to computing UB_1 we can compute UB_2
160     I_even1 = [];
```

```
161
162     %R(t_even) but replace UB_1(I) and LB_1(I)
163     % with UB_1^(1)(I), LB_1^(1)(I)
164     R_even1=[];
165     t_star1 = 2;
166
167     for t_even = 2:2:length(S)
168         R=[];
169
170         for t = t_even+1:2:length(S)
171
172             m_ub = []; %all m = product(I); |I| = t
173             m_lb = []; %all m = product(I); |I| = t-1
174
175             sum_ub = []; %sum over all |I| = t of UB_1(I)
176             sum_lb = []; %sum over all |I| = t-1 of LB_1(I)
177             for i = 1:length(I_factor)
178                 if I_factor(i) == t
179                     m_ub = [m_ub I(i)];
180
181                 elseif I_factor(i) == t-1
182                     m_lb = [m_lb I(i)];
183
184                 end
185             end
186
187             for j = 1:length(m_ub)
188                 if mod(m_ub(j),2) == 1
189                     m=m_ub(j);
190                     a=min((q^4-1)/(2*m)+(q-1)/2*q^2 ...
191                         -1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
192                 else
193                     m=m_ub(j);
194                     a=min(floor((q^4-2*q^2-3)/(2*m) ...
195                         +(q-1)/2*q^2 ...
196                         -1/2*gcd(q-1,(q^4-1)/m)),(q^4-1)/m);
197                 end
198                 sum_ub = [sum_ub a];
199             end
200
201             for j = 1:length(m_lb)
202                 if mod(m_lb(j),2) == 1
203                     m=m_lb(j);
204                     b = max((q^4-1)/(2*m)-(q-1)/2*q^2 ...
205                         -1/2*gcd(q-1,(q^4-1)/m),0);
206                 else
207                     m=m_lb(j);
```

```
208                     b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
209                         -(q-1)/2*q^2 ...
210                         -1/2*gcd(q-1,(q^4-1)/m),0);
211                 end
212                 sum_lb = [sum_lb b];
213             end
214
215             R_ub = sum(sum_ub);
216             R_lb = sum(sum_lb);
217
218             R = [R R_ub-R_lb];
219         end
220         d=sum(R);
221         R_even1 = [R_even1 d];
222         if d>0
223             I_even1 = [I_even1 t_even];
224         end
225     end
226
227     %Now we have R(t_even)(1)
228     %I_even1 consists of t_even which are
229     % strictly positive with R(t_even)(1)
230
231     %now we want to find t_star1
232     %t_star s.t. R(t_star)(1) > R(t_even)(1)
233     % for all t_even
234
235     R_max = max(R_even1);
236     counter = 1;
237
238     while R_even1(counter) ~= R_max
239         t_star1 = I_even1(counter+1);
240         counter = counter + 1;
241     end
242
243
244     %Now we have all we need to calculate UB_2
245
246     %Assume I_even not empty
247
248     UB_2 = 0;
249
250     for t=1:2:t_star1-1
251
252         m_ub = [];
253         m_lb = [];
254
```

```
255         sum_ub = [];
256         sum_lb = [];
257
258         for i=1:length(I_factor)
259             if  I_factor(i) == t
260                 m_ub = [m_ub I(i)];
261             elseif I_factor(i) == t-1
262                 m_lb = [m_lb I(i)];
263             end
264         end
265
266         for j=1:length(m_ub)
267
268             if mod(m_ub(j),2)==1
269                 m = m_ub(j);
270                 a=min((q^4-1)/(2*m)+(q-1)/2*q^2 ...
271                     -1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
272             else
273                 m = m_ub(j);
274                 a=min(floor((q^4-2*q^2-3)/(2*m)) ...
275                     +(q-1)/2*q^2 ...
276                     -1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
277             end
278             sum_ub = [sum_ub a];
279         end
280
281         for j=1:length(m_lb)
282
283             if mod(m_lb(j),2)==1
284                 m = m_lb(j);
285                 b = max((q^4-1)/(2*m) ...
286                     -(q-1)/2*q^2 ...
287                     -1/2*gcd(q-1,(q^4-1)/m),0);
288             else
289                 m = m_lb(j);
290                 b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
291                     -(q-1)/2*q^2 ...
292                     -1/2*gcd(q-1,(q^4-1)/m),0);
293             end
294             sum_lb = [sum_lb b];
295         end
296
297         UB_2 = UB_2 + sum(sum_ub)-sum(sum_lb);
298
299     end
300
301     UpperBound = (q+1)*eulerPhi(q^4-1) ...
```

```
302     -(q^4-1)*(q-3)/2-q*(q-1)*(q^2+1)+(q-3)*UB_1+2*UB_2;
303 end
```

```
1  function [LowerBound] = LB_New(q)
2      S=factor(q^4-1);
3
4      %S=[a a a b b c ...]
5
6      n=length(S);
7
8      I=divisors(q^4-1);
9      I=I(1,2:end);
10
11
12     %I_factor = |I| for all I in S
13     I_factor = zeros(1,length(I));
14     for i = 1:length(I)
15         m_ub = length(factor(I(i)));
16         I_factor(i)=m_ub;
17     end
18
19     %We want to find I_odd
20     %I_odd consists of integers t_odd s.t. ;
21     %3 <= t_odd <= |S|, L(t_odd) < 0
22
23     I_odd = [];
24     L_odd=[]; %R(t_odd)
25     t_star = 3;
26
27     for t_odd = 3:2:length(S)
28         L=[];
29
30
31         for t = t_odd:2:length(S)
32
33
34             m_ub = []; %all m = product(I); |I| = t
35             m_lb = []; %all m = product(I); |I| = t+1
36
37             sum_ub = []; %sum over all |I| = t of UB_1(I)
38             sum_lb = []; %sum over all |I| = t+1 of LB_1(I)
39             for i = 1:length(I_factor)
40                 if I_factor(i) == t+1
41                     m_ub = [m_ub I(i)];
42
43                 elseif I_factor(i) == t
44                     m_lb = [m_lb I(i)];
```

```
45
46                  end
47              end
48
49              for j = 1:length(m_ub)
50                  if mod(m_ub(j),2) == 1
51                      m=m_ub(j);
52                      a=min((q^4-1)/(2*m)+(q-1)/2*q^2, ...
53                          (q^4-1)/m);
54                  else
55                      m=m_ub(j);
56                      a=min(floor((q^4-2*q^2-3)/(2*m)) ...
57                          +(q-1)/2*q^2,(q^4-1)/m);
58                  end
59                  sum_ub = [sum_ub a];
60              end
61
62              for j = 1:length(m_lb)
63                  if mod(m_lb(j),2) == 1
64                      m=m_lb(j);
65                      b = max((q^4-1)/(2*m)-(q-1)/2*q^2,0);
66                  else
67                      m=m_lb(j);
68                      b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
69                          -(q-1)/2*q^2,0);
70                  end
71                  sum_lb = [sum_lb b];
72              end
73
74              L_ub = sum(sum_ub);
75              L_lb = sum(sum_lb);
76
77              L = [L L_lb-L_ub];
78          end
79          d=sum(L);
80          L_odd = [L_odd d];
81          if d<0
82              I_odd = [I_odd t_odd];
83          end
84      end
85
86      %Now we have L(t_odd)
87      %I_odd consists of t_odd which are strictly negative in L()
88
89      %now we want to find t_star
90      %t_star s.t. L(t_star) < L(t_odd) for all t_odd
91
```

```
92
93      L_min = min(L_odd);
94      counter = 1;
95
96      while L_odd(counter) ~= L_min
97          t_star = I_odd(counter+1);
98          counter = counter + 1;
99      end
100
101
102     %Now we have all we need to calculate LB_1
103
104
105     LB_1 = 0;
106
107     if isempty(I_odd) || length(S) < 3
108
109         for t=1:2:length(S)
110
111             m_ub = [];
112             m_lb = [];
113
114             sum_ub = [];
115             sum_lb = [];
116
117             for i=1:length(I_factor)
118                 if  I_factor(i) == t+1
119                     m_ub = [m_ub I(i)];
120                 elseif I_factor(i) == t
121                     m_lb = [m_lb I(i)];
122                 end
123             end
124
125             for j=1:length(m_ub)
126                 %check if m even or odd
127                 if mod(m_ub(j),2)==1
128                     m = m_ub(j);
129                     a=min((q^4-1)/(2*m)+(q-1)/2*q^2, ...
130                         (q^4-1)/m);
131                 else
132                     m = m_ub(j);
133                     a=min(floor((q^4-2*q^2-3)/(2*m)) ...
134                         +(q-1)/2*q^2,(q^4-1)/m);
135                 end
136                 sum_ub = [sum_ub a];
137             end
138
```

```
139             for j=1:length(m_lb)
140
141                 if mod(m_lb(j),2)==1
142                     m = m_lb(j);
143                     b = max((q^4-1)/(2*m)  ...
144                         -(q-1)/2*q^2,0);
145                 else
146                     m = m_lb(j);
147                     b = max(ceil((q^4+2*q^2-3)/(2*m))  ...
148                         -(q-1)/2*q^2,0);
149                 end
150                 sum_lb = [sum_lb b];
151             end
152
153             LB_1 = LB_1 + sum(sum_lb)-sum(sum_ub);
154         end
155
156     else
157         for t=1:2:t_star-2
158
159             m_ub = [];
160             m_lb = [];
161
162             sum_ub = [];
163             sum_lb = [];
164
165             for i=1:length(I_factor)
166                 if  I_factor(i) == t+1
167                     m_ub = [m_ub I(i)];
168                 elseif I_factor(i) == t
169                     m_lb = [m_lb I(i)];
170                 end
171             end
172
173             for j=1:length(m_ub)
174                 %check if m even or odd
175                 if mod(m_ub(j),2)==1
176                     m = m_ub(j);
177                     a=min((q^4-1)/(2*m)  ...
178                         +(q-1)/2*q^2,(q^4-1)/m);
179                 else
180                     m = m_ub(j);
181                     a=min(floor((q^4-2*q^2-3)/(2*m))  ...
182                         +(q-1)/2*q^2,(q^4-1)/m);
183                 end
184                 sum_ub = [sum_ub a];
185             end
```

```
186
187             for j=1:length(m_lb)
188
189                 if mod(m_lb(j),2)==1
190                     m = m_lb(j);
191                     b = max((q^4-1)/(2*m)  ...
192                         -(q-1)/2*q^2,0);
193                 else
194                     m = m_lb(j);
195                     b = max(ceil((q^4+2*q^2-3)/(2*m))  ...
196                         -(q-1)/2*q^2,0);
197                 end
198                 sum_lb = [sum_lb b];
199             end
200
201             LB_1 = LB_1 + sum(sum_lb)-sum(sum_ub);
202         end
203
204
205     end
206
207     LB_1=max(LB_1,0);
208     %Analog we can compute LB_2
209
210     I_odd1 = [];
211     L_odd1=[]; %L(t_odd)
212     t_star1 = 3;
213
214     for t_odd = 3:2:length(S)
215         L=[];
216
217
218         for t = t_odd:2:length(S)
219
220
221             m_ub = []; %all m = product(I); |I| = t
222             m_lb = []; %all m = product(I); |I| = t+1
223
224             sum_ub = []; %sum over all |I| = t of UB_1(I)
225             sum_lb = []; %sum over all |I| = t+1 of LB_1(I)
226             for i = 1:length(I_factor)
227                 if I_factor(i) == t+1
228                     m_ub = [m_ub I(i)];
229
230                 elseif I_factor(i) == t
231                     m_lb = [m_lb I(i)];
232
```

```
233                 end
234             end
235
236             for j = 1:length(m_ub)
237                 if mod(m_ub(j),2) == 1
238                     m=m_ub(j);
239                     a=min((q^4-1)/(2*m)+(q-1)/2*q^2 ...
240                         -1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
241                 else
242                     m=m_ub(j);
243                     a=min(floor((q^4-2*q^2-3)/(2*m)) ...
244                         +(q-1)/2*q^2-1/2*gcd(q-1,(q^4-1)/m), ...
245                         (q^4-1)/m);
246                 end
247                 sum_ub = [sum_ub a];
248             end
249
250             for j = 1:length(m_lb)
251                 if mod(m_lb(j),2) == 1
252                     m=m_lb(j);
253                     b = max((q^4-1)/(2*m)-(q-1)/2*q^2 ...
254                         -1/2*gcd(q-1,(q^4-1)/m),0);
255                 else
256                     m=m_lb(j);
257                     b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
258                         -(q-1)/2*q^2-1/2*gcd(q-1,(q^4-1)/m),0);
259                 end
260                 sum_lb = [sum_lb b];
261             end
262
263             L_ub = sum(sum_ub);
264             L_lb = sum(sum_lb);
265
266             L = [L L_lb-L_ub];
267         end
268         d=sum(L);
269         L_odd1 = [L_odd1 d];
270         if d<0
271             I_odd1 = [I_odd1 t_odd];
272         end
273     end
274
275     %Now we have L(t_odd)
276     %I_odd consists of t_odd which are strictly negative in L()
277
278     %now we want to find t_star
279     %t_star s.t. L(t_star) < L(t_odd) for all t_odd
```

```
280
281
282     L_min = min(L_odd1);
283     counter = 1;
284
285     while L_odd1(counter) ~= L_min
286         t_star1 = I_odd1(counter+1);
287         counter = counter + 1;
288     end
289
290
291     %Now we have all we need to calculate LB_2
292
293
294     LB_2 = 0;
295
296     if isempty(I_odd1) || length(S) < 3
297
298         for t=1:2:length(S)
299
300             m_ub = [];
301             m_lb = [];
302
303             sum_ub = [];
304             sum_lb = [];
305
306             for i=1:length(I_factor)
307                 if  I_factor(i) == t+1
308                     m_ub = [m_ub I(i)];
309                 elseif I_factor(i) == t
310                     m_lb = [m_lb I(i)];
311                 end
312             end
313
314             for j=1:length(m_ub)
315                 %check if m even or odd
316                 if mod(m_ub(j),2)==1
317                     m = m_ub(j);
318                     a=min((q^4-1)/(2*m)+(q-1)/2*q^2 ...
319                         -1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
320                 else
321                     m = m_ub(j);
322                     a=min(floor((q^4-2*q^2-3)/(2*m)) ...
323                         +(q-1)/2*q^2- ...
324                         1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
325                 end
326                 sum_ub = [sum_ub a];
```

```
327             end
328
329             for j=1:length(m_lb)
330
331                 if mod(m_lb(j),2)==1
332                     m = m_lb(j);
333                     b = max((q^4-1)/(2*m)-(q-1)/2*q^2 ...
334                         -1/2*gcd(q-1,(q^4-1)/m),0);
335                 else
336                     m = m_lb(j);
337                     b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
338                         -(q-1)/2*q^2- ...
339                         1/2*gcd(q-1,(q^4-1)/m),0);
340                 end
341                 sum_lb = [sum_lb b];
342             end
343
344             LB_2 = LB_2 + sum(sum_lb)-sum(sum_ub);
345         end
346
347     else
348         for t=1:2:t_star-2
349
350             m_ub = [];
351             m_lb = [];
352
353             sum_ub = [];
354             sum_lb = [];
355
356             for i=1:length(I_factor)
357                 if  I_factor(i) == t+1
358                     m_ub = [m_ub I(i)];
359                 elseif I_factor(i) == t
360                     m_lb = [m_lb I(i)];
361                 end
362             end
363
364             for j=1:length(m_ub)
365                 %check if m even or odd
366                 if mod(m_ub(j),2)==1
367                     m = m_ub(j);
368                     a=min((q^4-1)/(2*m)+(q-1)/2*q^2 ...
369                         -1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
370                 else
371                     m = m_ub(j);
372                     a=min(floor((q^4-2*q^2-3)/(2*m)) ...
373                         +(q-1)/2*q^2- ...
```

```
                    1/2*gcd(q-1,(q^4-1)/m),(q^4-1)/m);
                end
                sum_ub = [sum_ub a];
            end

            for j=1:length(m_lb)

                if mod(m_lb(j),2)==1
                    m = m_lb(j);
                    b = max((q^4-1)/(2*m)-(q-1)/2*q^2 ...
                        -1/2*gcd(q-1,(q^4-1)/m),0);
                else
                    m = m_lb(j);
                    b = max(ceil((q^4+2*q^2-3)/(2*m)) ...
                        -(q-1)/2*q^2 ...
                        -1/2*gcd(q-1,(q^4-1)/m),0);
                end
                sum_lb = [sum_lb b];
            end

            LB_2 = LB_2 + sum(sum_lb)-sum(sum_ub);
        end

        LB_2=max(LB_2,0);

    end
    LowerBound = (q+1)*eulerPhi(q^4-1) ...
    -(q^4-1)*(q-3)/2-q*(q-1)*(q^2+1)+(q-3)*LB_1+2*LB_2;
```

Bibliography

[1] R. A. Beeler. *How to Count: An Introduction to Combinatorics and Its Applications*. Springer, 2015.

[2] M. B. Can and F. Ozbudak. Bounding the sum of μ-invariants on pair symbol weights over some irreducible codes. *IFAC-PapersOnLine*, 58(17):298–303, 2024.

[3] Y. Cassuto and M. Blaum. Codes for symbol-pair read channels. In *2010 IEEE International Symposium on Information Theory*, pages 988–992. IEEE, 2010.

[4] Y. Cassuto and M. Blaum. Codes for symbol-pair read channels. *IEEE Transactions on Information Theory*, 57(12):8011–8020, 2011.

[5] Y. M. Chee, H. M. Kiah, and C. Wang. Maximum distance separable symbol-pair codes. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2886–2890. IEEE, 2012.

[6] B. Chen, L. Lin, and H. Liu. Constacyclic symbol-pair codes: Lower bounds and optimal constructions. *IEEE Transactions on Information Theory*, 63(12):7661–7666, 2017.

[7] B. Ding, G. Ge, J. Zhang, T. Zhang, and Y. Zhang. New constructions of MDS symbol-pair codes. *Designs, Codes and Cryptography*, 86:841–859, 2018.

[8] B. Ding, T. Zhang, and G. Ge. Maximum distance separable codes for b-symbol read channels. *Finite Fields and Their Applications*, 49:180–197, 2018.

[9] H. Freedman. Introduction to finite fields and their applications (revised edition), by Rudolf Lidl and Harald Niederreiter. (Cambridge University Press). *The Mathematical Gazette*, 79(485):439–439, 1995.

[10] X. Kai, S. Zhu, and P. Li. A construction of new MDS symbol-pair codes. *IEEE Transactions on Information Theory*, 61(11):5828–5834, 2015.

[11] S. Lang. *Algebra*, volume 211. Springer Science & Business Media, 2012.

[12] R. Lidl and H. Niederreiter. *Finite fields*. Number 20. Cambridge University Press, 1997.

[13] R. J. McEliece. *Finite fields for computer scientists and engineers*, volume 23. Springer Science & Business Media, 2012.

[14] M. J. Moisio. A note on evaluations of some exponential sums. *Acta Arithmetica*, 93(2):117–119, 2000.

[15] R. M. Roth. Introduction to coding theory. *IET Communications*, 47(18-19):4, 2006.

[16] B. Schmidt and C. White. All two-weight irreducible cyclic codes? *Finite Fields and Their Applications*, 8(1):1–17, 2002.

[17] M. Shi, F. Özbudak, and P. Solé. Geometric approach to b-symbol Hamming weights of cyclic codes. *IEEE Transactions on Information Theory*, 67(6):3735–3751, 2021.

[18] H. Stichtenoth. *Algebraic function fields and codes*, volume 254. Springer Science & Business Media, 2009.

[19] J. H. Van Lint. *Introduction to coding theory*, volume 86. Springer Science & Business Media, 1998.

[20] G. Vega. A characterization of all semiprimitive irreducible cyclic codes in terms of their lengths. *Applicable Algebra in Engineering, Communication and Computing*, 30(5):441–452, 2019.

[21] G. Vega. The b-symbol weight distributions of all semiprimitive irreducible cyclic codes. *Designs, Codes and Cryptography*, 91(6):2213–2221, 2023.

[22] E. Yaakobi, J. Bruck, and P. H. Siegel. Constructions and decoding of cyclic codes over b-symbol read channels. *IEEE Transactions on Information Theory*, 62(4):1541–1551, 2016.

[23] H. Zhu, M. Shi, and F. Özbudak. Complete b-symbol weight distribution of some irreducible cyclic codes. *Designs, Codes and Cryptography*, 90(5):1113–1125, 2022.