University of Puerto Rico
Río Piedras Campus
Faculty of Natural Sciences
Department of Mathematics

A GO-UP CONSTRUCTION AND APPLICATIONS

By

Eddie Arrieta Arrieta

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN MATHEMATICS AT THE UNIVERSITY OF PUERTO RICO, RÍO PIEDRAS CAMPUS

16 July, 2021

APPROVED BY THE DOCTORAL DISSERTATION
COMMITTEE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY IN MATHEMATICS
AT THE UNIVERSITY OF PUERTO RICO
RÍO PIEDRAS CAMPUS

ADVISOR:

Heeralal Janwa, Ph.D.
University of Puerto Rico, Río Piedras.

READERS:

Bud Mishra, Ph.D.
New York University

Harold F. Mattson Jr, Ph.D.
Syracuse University

Ivelisse Rubio Canabal, Ph.D.
University of Puerto Rico, Río Piedras.

Puhua Guan, Ph.D.
University of Puerto Rico, Río Piedras.

Abstract of the Ph.D Thesis Presented to the Graduate School
of the University of Puerto Rico, Río Piedras Campus in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy in Mathematics

A GO-UP CONSTRUCTION AND APPLICATIONS

By

Eddie Arrieta Arrieta

16 July 2021

Chair: Heeralal Janwa Ph.D.
Major Department: Mathematics

Given a code C over the finite field $\mathbb{F}_q$, where q is a power of a prime number, some constructions exist that permit us to obtain a new code from C over $\mathbb{F}_q$ or over a subfield of $\mathbb{F}_q$, for example as subfield subcodes. However, in some important applications, one needs codes over an extension field, such as quantum error-correcting codes (QECC). In these applications, the codes in the extension fields need only be additive. In this dissertation, we propose a novel technique that we call **Go-Up** (**GU**) construction that allows us to obtain an additive or a linear code over $\mathbb{F}_{q^m}$ from a collection of codes over $\mathbb{F}_q$. We show under what condition is this code a self-orthogonal or self-dual code, under various forms (Euclidean, or Hermitian, or Trace Hermitian, or alternating). We use these additive self-orthogonal codes to construct quantum-stabilizer codes. As a result, we give explicit classes of codes where we obtain new quantum error-correcting codes.

Under certain conditions, we show that the **GU** of two classical Goppa codes is a Goppa code. As a consequence, we obtain quantum error-correcting codes from classical Goppa codes. So far, research on QECC from Goppa codes has been

limited. Also, the decoding of QECC is limited. Our result might allow fast decoding of QECC from **GU** of Goppa codes via Patterson's $O(n, log\ n)$ decoding algorithm.

Independent of the **GU** construction, we show under what conditions are classical Goppa code self-orthogonal. Using this fact, we obtain binary and quaternary QECC directly from Goppa codes. We compare our results on QECC to those obtained for QECC from BCH codes by Beth and Grassl, and by Sarvepalli et al.

The third theme of our thesis is to construct few weights codes. Such codes have applications in cryptography, association schemes, Steiner systems, t-designs, strongly regular graphs, finite group theory, finite geometries, among other disciplines. The theory and construction of two-weight linear codes have been carried out by Calderbank and Kantor, among others. Tonchev and Jungnickel, and Ding et al. have done pioneering work on three-weight codes. We use our **GU** code construction to obtain two-weight, three-weight and few-weights linear codes. Consequently, we give elementary construction of class RT1 of two-weight codes by Calderbank and Kantor. We also obtain new classes of three-weight codes.

To My Parents, to Geli and to Mompox, the place that does not exist.

ACKNOWLEDGMENTS

I feel I must express, as a person and as a student, my gratitude to many who have directly or indirectly contributed to bringing this dissertation into shape. First of all, I would like to gratefully acknowledge my advisor, Dr. Heeralal Janwa for providing me with a lot of comments and advice, reading each original report, pointing out weaknesses, and suggesting improvements.

I have special gratitude towards the Program of Formative Academic Experiences, and the Dean of Graduate Studies and Investigation for the financial support.

This dissertation would not have been possible without the undiminishing support from my parents, to whom I am deeply grateful for their faith and prayers.

On a personal note, I thank Mr. Ambassador Carlos A Forero for his comments on English grammar.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

Symbol	Meaning	See
$\|\boldsymbol{v}>^*$	The adjoint of a ket, called bra	Equation (2.2)
σ_x, σ_z	Bit and phase error Pauli matrix	Table 2–4
$\mathbf{\Gamma}(L,\ g(x))$	Goppa code with support L and polynomial $g(x)$	Definition 2.7
$(A, B)_F$	Frobenius inner product of matrices A and B	Equation (2.9)
$\mathbb{H}$	Hilbert space	Definition 2.9
U	Unitary operator or quantum evolution	Definition 2.10
Q	Quantum error-correcting code	Definition 2.12
$\mathbf{B}_n(q)$	Nice Error Basis in n qubits	Equation (2.14)
ξ_n	The error group associated with the nice error basis $\mathbf{B}_n(q)$	Equation (2.16)
$[[n,\ k,\ d]]_q$	Quantum stabilizer code with dimension q^k and minimum distance d over $\mathbb{F}_q$	Definition 2.16
$\mathbf{GU}(C_0, \cdots, C_{m-1})$	amalgamation of the m linear codes C_i	Definition 3.1
$\mathbf{GU}(m, C_0)$	Self-amalgamation of m copies of C_0	Definition 3.1
$\hat{\mathbf{\Gamma}}(L, g)$	Self-amalgamated Goppa code	Theorem 3.2
$S_k(q)$	The general Simplex code	Lemma 3.4

CHAPTER 1
INTRODUCTION

Quantum error-correcting codes are a key ingredient to implement information processing based on quantum mechanic [13, 14, 25, 31]. In the late $1990's$ Gottesman [25] and independently Calderbank et al., [14] proposed a method to construct quantum codes from classical codes. This method is commonly referred to a *stabilizer codes*. Theses codes are the most studied class of quantum codes together with the so called CSS construction introduced by Shor et al., [15] and independently by Steane [48]. In this thesis, we give a new construction of additive codes and, from them, we construct quantum stabilizer codes.

Another important problem, still open, in quantum error-correcting codes is to give fast decoding algorithms for error-correction. We give here, quantum stabilizer code construction using Goppa codes. Since Goppa codes have a fast decoding algorithm by Petterson [40], our quantum stabilizer codes have a promise of fast decoding.

This thesis also addresses with few weights codes, they are interesting in classical coding theory and have application to finite group theory, finite geometry, cryptography and other research areas in discrete mathematics. In particular, two and three-weight codes are closely related to strongly regular graphs, secret sharing schemes, and projective point sets. Delsarte was the first to investigate the relationship between projective sets, graphs and linear codes in the early $1970's$ [16, 18–20, 22, 49]. Two-weight codes can be used to construct secret sharing schemes, an interesting subject in cryptography introduced by Blackley [10] and Shamir [46].

This dissertation presents a systematic method to construct few-weights codes (linear and additive) over a finite extension of the finite field $\mathbb{F}_q$. The theory of additive codes gives rich dividends, for example in the construction of quantum stabilizer codes [14, 23, 32, 51]. We get quaternary additive codes with optimal parameters, the quaternary case is of special interest because of a close link to the theory of quantum stabilizer codes [23, 51]. In addition, we get optimal parameters for single quantum state codes, such a code might be useful for example in testing whether certain storage locations for qubits are decohering faster than they should [14].

Another important problem is to be able to construct quantum Stabilizer codes solely based on Euclidean self-orthogonality (most of the constructions need to have Trance Hermitian orthogonality or alternating duality conditions). From our method, we are able to construct stabilizer codes using Euclidean self-orthogonal codes and for the first time we use Goppa code to construct stabilizer codes. These codes could, for the first time, yield fast decodable quantum stabilizer codes using Patterson's $O(n \log n)$ algorithm for the classical Goppa codes.

Given a code C over the finite field $\mathbb{F}_q$, where q is a power of a prime number, some constructions exist that permit us to obtain a new code from C **over** $\mathbb{F}_q$ or **over a subfield of** $\mathbb{F}_q$, such as subfield subcodes. However, in some important applications, one needs codes over an extension field, for example, in quantum error-correcting codes. We present a new and elegant technique that we call **Go-Up** construction, which allows us to obtain an additive or a linear code over $\mathbb{F}_{q^m}$ from any set of m linear codes over $\mathbb{F}_q$.

We have found a nice relation between our **Go-Up** construction and important topics in coding theory such as:

a. Binary and nonbinary quantum error correcting codes.

b. Two, three and few-weights codes.

c. Frobenius invariant codes.

In fact, the initial code C could have some particular properties, namely:

- If $C = \Gamma(g(x),\ L)$ is a $q-$ary Goppa code, we study the properties of the go-up built code, getting - under some conditions - a code that is also a Goppa code over $\mathbb{F}_{q^2}$.
- If C is constant weight, as the simplex code, then the new one is two-weight linear code, see Theorem 3.6. Table 3–9 shows a family of two-weight codes over $\mathbb{F}_4$. Comparing our technique with the construction given by Calderbank et al., in [16], where they get a new two-weight code from a given two-weight code by changing the underlying field, we observe that our method uses elementary tools to obtain two-weight codes. For example, when the initial code is the general simplex code $S_k(q)$, we get a two-weight code with the same parameters as the code obtained in *Example RT1* from [16].
- If C is a binary two-weight code, we apply our construction to it and we obtain a linear code over $\mathbb{F}_4$, which is almost a six-weight code, see Theorem 3.7. For example, if C is the first-order Reed-Muller code, the new one is just three-weight, and Table 3–10 shows us its weight distribution. We observe that our construction permits us to get examples of three-weight codes in an easier way comparing with the method used by K. Ding et al., in [22].
- If C is a $q-$ary antipodal two-weight linear code, see Theorem 2.7, and applying to C our construction, we obtain a three-weight linear code over $\mathbb{F}_{q^2}$. Theorems 3.8 and 3.9 give the weight distribution of the new code.
- If C is a binary Euclidean self-orthogonal code, we construct a stabilizer group of unitary operators from the built code. We use a polynomial basis of $\mathbb{F}_4$ over $\mathbb{F}_2$ obtaining as a result that our new code is Euclidean self-orthogonal as C is, in contrast with the seminal work [14] where they use self-orthogonality with respect to the Trace Hermitian inner product and a normal basis.

- If C is a Euclidean self-orthogonal code over $\mathbb{F}_q$, we found that the Euclidean duality for our new code over $\mathbb{F}_{q^2}$ is the same as the trace-alternating duality used in the work [31]. Then, we can obtain a nonbinary stabilizer code without the need to verify alternating self-orthogonality of the built code from C. We just need to begin with classical self-orthogonality and many types of codes are classically self-orthogonal.

The new code, in general, is obtained from m codes in $\mathbb{F}_q^n$, and using a polynomial or normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Indeed, we can use any basis, always resulting in an additive code, see Definitions 3.1 and 3.2. When the m codes are linear, the resulting new code is linear over $\mathbb{F}_{q^m}$ if and only if they are equal, see Lemma 3.1. In the linear and binary case, if $m = 2$ and C_0 is Euclidean self-orthogonal or self-dual, the generated code is also Euclidean self-orthogonal or self-dual, with the additional property that Euclidean duality is equivalent with the Trace Euclidean, Hermitian, and Trace Hermitian duality. See Proposition 3.7.

In the linear and $q-$ary scenario with $q - 3$ divisible by 4, for example the Mersenne prime numbers, if $m = 2$ we get the same equivalence among the dualities, being this last fact a general version of the Theorem 3 from work [14], see Theorem 3.5.

On the other hand, if we take a code C over $\mathbb{F}_{q^m}$ that is Frobenius invariant, that is, for any codeword $\boldsymbol{x} = (x_1 \cdots x_n)$ in C we have $\boldsymbol{x}^q = (x_1^q \cdots x_n^q)$ is also a codeword of C, and C_0 being its subfield subcode over $\mathbb{F}_q$, the code C can be obtained from C_0 through our **Go-Up** construction using the Delsarte's theorem, as it is proved in Lemma 3.2 and Theorem 3.1.

This dissertation is structured as follows: **Chapter 2** introduces some basic concepts about coding theory; the principal references are the books [6, 7, 9, 38, 39, 41, 50]. We begin with some definitions and theorems about error-correcting codes.

In the first section, we introduce some basic facts about projective geometry and two-weight codes. It ends with Theorem 2.8 which gives a relation between projective sets and two-weight codes. **Section 2.2** starts with basic properties of Goppa codes and their relation with BCH codes. This section ends with Proposition 2.3, where we got a lower bound for the minimum distance of a binary Goppa code when its polynomial has the form $g(x) = x^l h(x)$ over $\mathbb{F}_{2^m}$ and $h(x)$ being separable such that $h(0) \neq 0$. **Section 2.3** introduces the basics of quantum stabilizer codes. We have introduced Hilbert spaces, facts about simultaneously diagonalizable operators, and stabilizer codes, see Definition 2.13. In the end, we define the concepts of *nice error basis* and the *error group* associated with the error basis. In this section the principal references are [1, 3, 5, 13, 24, 25, 31, 33, 35, 42, 47], the books [27, 37, 43, 44] and the seminal works [14, 15].

Chapter 3 introduces our **Go-Up** construction, we call it amalgamation. We obtain Lemma 3.1, showing that, in general, an amalgamated code is not linear over $\mathbb{F}_{q^m}$. We have found an equivalence between a Frobenius invariant linear code over $\mathbb{F}_{q^m}$ and a code obtained from our construction. Theorem 3.1 shows such equivalence. **Section 3.2** is devoted to study the Goppa codes to which we apply our construction, getting Theorem 3.3, which shows that the amalgamation of a Goppa code is also a Goppa code. **Section 3.3** is dedicated to study the dual of the amalgamated code. We obtain Theorem 3.4 and Proposition 3.6. In **Section 3.4** we generalize Proposition 3.6 considering a prime $q \equiv 3 \pmod 4$. Proposition 3.8 shows such generalization in this $q-$ary case. Theorem 3.5 is a general version of the *Theorem* 3 of [14].

Section 3.5 contains the main results of Chapter 3. We discuss various applications. We construct two and three-weight linear or additive codes. First, we obtain Theorem 3.6, where we begin with a constant weight binary code, and we apply to it our construction. Second, applying our technique to a binary two-weight

code C, we get a code over $\mathbb{F}_4$ which is almost six-weight, see Theorem 3.7. When C is the first-order generalized Reed-Muller code, we get a three-weight code over $\mathbb{F}_{q^2}$, see Theorem 3.8. We also obtain three-weight and few-weight additive codes, see Theorem 3.10.

Chapter 4 is devoted to the study of stabilizer codes obtained from our construction. In the first section, given an amalgamated code over $\mathbb{F}_4$, which is Euclidean self-orthogonal, we get a quantum stabilizer code (see Proposition 4.1 and Theorem 4.1).

Section 4.2 discusses that for an amalgamated code C in $\mathbb{F}_{q^2}^n$, the Euclidean duality is the same as the trace-alternating duality considered in [31]. That allows us to construct nonbinary stabilizer codes taking into account Euclidean duality and Frobenius invariant codes, see Proposition 4.4 and Theorem 4.2. We give another way to construct q-ary **QECC**. Calderbankd and Shor [14] obtain **QECC** from binary quantum codes by showing that under their conditions, Hermitian and Trace Hermitian inner products are equivalent. Using our **Go-Up** construction, we are able to obtain the same equivalence of these inner products for q-ary codes when q is a prime such that $q \equiv 3 \pmod 4$, using Theorem 3.5. Therefore with our Go-Up construction, we can obtain q-ary **QECC**.

Chapters 3 and 4 contain our main contributions. Some preliminary results are presented on separable and inseparable Goppa codes in Chapter 2, see Proposition 2.3. The following table summarizes the main results in this dissertation.

The main contributions	
Theorem 3.1	$C \subset \mathbb{F}_{q^m}^n$ and $C^q \subset C$, C is linear over $\mathbb{F}_{q^m}$ if and only if $C = \mathbf{GU}(m,\ C_0)$, C_0 is the sub-field sub-code of C over $\mathbb{F}_q$.
Theorem 3.4	Let $\mathbb{F}_q$ be a finite field of characteristic two. Take C_0 and C_1 linear codes over $\mathbb{F}_q$, with $C_0 \subset C_1$. The **additive code** $C = C_0 + \delta C_1^{\perp} \subset \mathbb{F}_{q^2}^n$ is Trace Hermitian self-orthogonal.
Theorem 3.5	Take a prime $q \equiv 3 \pmod 4$ and C a q^2-linear code. Then $C \subset C^{\perp_H}$ if and only if $C \subset C^{\perp_{TH}}$
Theorem 3.6	$C_0 \subset \mathbb{F}_q^n$ constant weight and **constant intersection** code, Definition 3.3. Then $\mathbf{GU}(2,\ C_0)$ is two-weight.
Theorem 3.8	$\mathbf{GU}(2,\ \boldsymbol{R}_q(1,k)) \subset \mathbb{F}_{q^2}^{q^k}$ is a three-weight linear code. $\boldsymbol{R}_q(1,k)$ is the first-order generalized Reed-Muller code over $\mathbb{F}_q$.
Theorem 3.10	$\mathbf{GU}(\boldsymbol{R}_q(1,k), C_0) \subset \mathbb{F}_{q^2}^{q^k}$ is a three-weight additive code, where $C_0 = \{(\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0) \mid \boldsymbol{a} \in S_k(q)\}$.
Theorem 4.2	C an $[n,k,d]_{q^2}$ Euclidean self-orthogonal and $C^q \subset C$. Then C yields an $[[n, n-2k, \geq d^{\perp}]]_q$ quantum stabilizer code.

Table 1–1

CHAPTER 2
BACKGROUND AND PRELIMINARY RESULTS

In this chapter, we present basic definitions and known results about general and quantum stabilizer codes that we will use in the subsequent chapters.

Section 2.1 begins with some definitions and theorems about error-correcting codes. It ends with some facts about projective geometry and its relationship with two-weight codes. **Section 2.2** starts with basic properties of Goppa codes and their relationship with BCH codes. This section ends with Proposition 2.3, where we got a lower bound for the minimum distance of a binary Goppa code when its polynomial has the form $g(x) = x^l h(x)$. **Section 2.3** introduces the basics of quantum stabilizer codes. We have introduced Hilbert spaces, facts about simultaneously diagonalizable operators, and stabilizer codes. In the end, we define the concepts of *nice error basis* and the *error group* associated with the basis error ξ_n.

2.1 BASIC DEFINITIONS

We give some general facts about coding theory. The principal references for this section are the books [6, 7, 9, 38, 39, 41, 50] and the works [11, 16, 22, 53].

A sender starts with a message and **encodes** it to obtain codewords consisting of sequences of symbols. These are transmitted over a noisy channel to the receiver. Often the sequence of symbols that are received contains errors and therefore might not be codewords. The receiver must **decode**, which means correct the errors, in order to change what is received back to codeword and then recover the original

message, that is, we can encrypt encoding and adding errors. After that, we can decrypt correcting the errors.

The symbols used to construct the codewords belong to an alphabet. A code that uses an alphabet consisting of q symbols is called $q-$ary code. If $\mathbb{A}$ is an alphabet, a code is a random subset of $\mathbb{A}^n$ where n is called the length of the code. In general, decoding (decrypting) could be a time-consuming procedure, therefore, the most useful codes are subsets of $\mathbb{A}^n$ satisfying additional conditions. The most common is to require $\mathbb{A}$ to be a finite field, $\mathbb{F}_q$, so that $\mathbb{A}^n$ is a vector space of dimension n, and require the code C to be a subspace of this vector space. In this case, we call k the dimension of C over $\mathbb{F}_q$.

To decode, we need to put a measure on how close two vectors are to each other. This measure is given by the Hamming distance. Let $\boldsymbol{u}$, $\boldsymbol{v}$ be two vectors in $\mathbb{A}^n$, the Hamming distance $d(\boldsymbol{u}, \boldsymbol{v})$ is the number of places where the two vectors differ. For example, if we use binary vectors and we have $\boldsymbol{u} = (1\ 0\ 1\ 0\ 1\ 0\ 1\ 0)$ and $\boldsymbol{v} = (1\ 0\ 1\ 1\ 1\ 0\ 0\ 0)$,

$$d(\boldsymbol{u}, \boldsymbol{v}) = 4.$$

As another example, suppose we are working with the usual English alphabet, then

$$d(\mathbf{decode}, \mathbf{vector}) = 4.$$

The importance of the Hamming distance $d(\boldsymbol{u}, \boldsymbol{v})$ is that it measures the minimum number of "errors" needed for $\boldsymbol{u}$ to be changed to $\boldsymbol{v}$.

The minimum distance of a code C is defined by

$$d = min\{d(\boldsymbol{u}, \boldsymbol{v}) : \boldsymbol{u}, \boldsymbol{v} \in C, \boldsymbol{u} \neq \boldsymbol{v}\}.$$

If $C \subset \mathbb{F}_q^n$ is a subspace, for $\boldsymbol{u}$ and $\boldsymbol{v}$ in C, $\boldsymbol{u} - \boldsymbol{v} = \boldsymbol{c} \in C$. Then we get that

$$d = min\{d(\mathbf{0}, \boldsymbol{c}) : \boldsymbol{c} \in C, \boldsymbol{c} \neq \mathbf{0}\}.$$

This number is very important because it gives the smallest number of errors needed to change one codeword into another. If a codeword is transmitted over a noisy channel, errors are introduced into some of the entries of the vector. *We can correct these errors by finding the codeword whose Hamming distance from the received vector is as small as possible.* This is called **nearest neighbor decoding**.

We say that a code can detect up to t errors if changing a codeword in at most t places can not change it to another codeword, that is, given a codeword $\boldsymbol{u}$ we have

$$\overline{B}(\boldsymbol{u}, t) \cap C = \emptyset,$$

where $\overline{B}(\boldsymbol{u}, t) = \{\boldsymbol{v} \in \mathbb{F}_q^n : d(\boldsymbol{u}, \boldsymbol{v}) \leq t\}$.

Theorem 2.1. *A code C can detect up to t errors if*

$$d \geq t + 1$$

and a code C can correct up to t errors if

$$d \geq 2t + 1$$

Proof. Suppose that $d \geq t + 1$, if a codeword $\boldsymbol{u}$ is sent and t errors or fewer occur, then the received message $\boldsymbol{v}$ can not be a different codeword because

$$d(\boldsymbol{u}, \boldsymbol{v}) \leq t < t + 1$$

then $\boldsymbol{v}$ is not an element of C.

Suppose that $d \geq 2t+1$. Assume that the codeword $\boldsymbol{u}$ is sent and the received word $\boldsymbol{v}$ has t or fewer errors, that is, $d(\boldsymbol{u}, \boldsymbol{v}) \leq t$. Let $\boldsymbol{x} \in C - \{\boldsymbol{u}\}$, then $d(\boldsymbol{x}, \boldsymbol{v}) \geq t + 1$ because if $d(\boldsymbol{x}, \boldsymbol{v}) \leq t$ we obtain

$$t + t \geq d(\boldsymbol{u}, \boldsymbol{v}) + d(\boldsymbol{v}, \boldsymbol{x}) \geq d(\boldsymbol{u}, \boldsymbol{x}) \geq 2t + 1$$

That is, $2t \geq 2t+1$ a contradiction. Thus $d(\boldsymbol{v}, \boldsymbol{x}) \geq t+1$, i.e. decoding $\boldsymbol{v}$ we obtain $\boldsymbol{u}$ by nearest neighbor decoding.

□

We observe that one way to find the nearest neighbor is to calculate the distance between the received message and each of the codewords, then select the codeword with the smallest Hamming distance. In practice, this is impractical for large codes. In general, the decoding problem is challenging, and considerable research effort is devoted to looking for fast decoding algorithms.

A q-ary code C of length n, with M codewords and minimum distance d is called an $(n,\ M,\ d)_q$ code. If C is a linear code of dimension k, we write $[n,\ k,\ d]_q$.

Definition 2.1. *If we have an* $(n,\ M,\ d)_q$ *code, then we define the information rate* R *as*

$$R = \frac{\log_q(M)}{n}.$$

For example, if C is a binary linear code of dimension k, we obtain $M = 2^k$ and then

$$R = \frac{\log_2(2^k)}{n} = \frac{k}{n}.$$

The code rate R represents the ratio of the number of input data symbols to the number of transmitted code symbols. It is an important parameter to consider when implementing real-world systems, as it represents what fraction of the bandwidth is being used to transmit actual data. We would like for M, for q^k in the linear case, to be large so that the code rate, R, will be as close to one as possible. This optimization allows us to use bandwidth efficiently when transmitting messages over a noisy channel. Unfortunately, increasing d tends to increase n or decrease M.

Theorem 2.2 (R. Singleton 1964)**.** *Let* C *be an* $(n,\ M,\ d)_q$ *code. Then*

$$M \leq q^{n+1-d}.$$

If C is an $[n,\ k,\ d]_q$ code, $q^k \leq q^{n+1-d}$, i.e, $k \leq n+1-d$.

Proof. For a codeword $\boldsymbol{a} = (a_1\ a_2 \cdots a_n)$, let $\boldsymbol{a}' = (a_d\ a_{d+1} \cdots a_n)$. If $\boldsymbol{b} \neq \boldsymbol{c}$ are two codewords then they differ in at least d places. Since $\boldsymbol{b}'$ and $\boldsymbol{c}'$ are obtained by removing $d-1$ entries from $\boldsymbol{b}$ and $\boldsymbol{c}$, they must differ in at least one place. So $\boldsymbol{b}' \neq \boldsymbol{c}'$. Therefore, the number M of codewords $\boldsymbol{u}$ equals the number of vectors $\boldsymbol{u}'$ obtained in this way. There are at most $q^{n-(d-1)}$ vectors $\boldsymbol{u}'$ since there are $n-(d-1)$ positions in these vectors. This implies that

$$M \leq q^{n-(d-1)} = q^{n+1-d}$$

as required. □

a. The code rate R of an $(n,\ M,\ d)_q$ code is at most $1-\frac{d-1}{n}$. Since we have $R = \frac{\log_q(M)}{n}$ and $M \leq q^{n+1-d}$, then $\log_q(M) \leq \log_q(q^{n+1-d}) = n+1-d$; that is,

$$R \leq \frac{n+1-d}{n} = 1 - \frac{d-1}{n}.$$

b. If we take the number $\frac{d}{n}$, called the **relative minimum distance**, the above observation implies that if $\frac{d}{n}$ is large, the code rate R, is forced to be small.

c. In the linear case of a code C of dimension k: For given n and d, the larger the k, the better the code, because we may think of each codeword as having k information symbols and $n-k$ checks. So, large k with respect to n makes an efficient code.

d. Given a code C of minimum distance d: the larger, the better because we can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.

e. The Singleton bound can be rewritten as

$$q^d \leq \frac{q^{n+1}}{M}$$

or

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

in the linear case with dimension k.

Definition 2.2. *A code that satisfies the Singleton bound with equality is called MDS code (Maximum Distance Separable).*

a. We observe that if C is an MDS code, C has the largest possible value of d for a given M and n, in the linear case $d = n + 1 - k$; and, if C is an MDS code,

$$R = 1 - \frac{d-1}{n}.$$

b. The only binary MDS codes are the trivial codes, i.e. $[n,\ 1,\ n]_2$, the repetition code, $[n,\ n-1,\ ,2]_2$, the parity check code and $[n,\ n,\ 1]_2$, the universal code.

In fact:

Let $C \subset \mathbb{F}_2^n$ be an $MDS-$code with $G = [I_k|A]$ a generator matrix, where $A \in \mathbb{M}_{k\times(n-k)}(\mathbb{F}_2)$. Since $d = (n-k)+1$ all the entries of A are not zero, i.e. all the entries of A are ones. If A has at least two rows, taking $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, the first two rows of G, $\boldsymbol{r}_1+\boldsymbol{r}_2 \in C$ and $\omega(\boldsymbol{r}_1+\boldsymbol{r}_2) = 2$ because $\boldsymbol{r}_1+\boldsymbol{r}_2 = (1\ 0\ \cdots\ 0\ 1\ 1\ \cdots\ 1)+(0\ 1\ \cdots\ 0\ 1\ 1\ \cdots\ 1) = (1\ 1\ 0\ \cdots\ 0)$. Then $d \leq 2$, $0 \leq n-k+1 \leq 2$. That is, $n = k$ or $n = k+1$. Thus, C has parameters $[n,\ n,\ 1]_2$ or $[n,\ n-1,\ 2]_2$. If A has one row, $k = 1$, $d = n-1+1 = n$, and the code has parameters $[n,\ 1,\ n]_2$.

The following is a geometric interpretation that is useful in error correcting. A Hamming sphere of radius t centered at a codeword $\boldsymbol{c}$ is denoted by $B(\boldsymbol{c},t)$ and defined by

$$B(\boldsymbol{c},t) = \{\boldsymbol{u} \in \mathbb{F}_q^n : d(\boldsymbol{c},\boldsymbol{u}) \leq t\} = \overline{B}(\boldsymbol{c},t).$$

We calculate the number of vectors in $B(\boldsymbol{c},t)$.

Lemma 2.1. *Let C be an $(n,\ M,\ d)_q$ code. Then*

$$|B(\boldsymbol{c},t)| = \sum_{i=0}^{t} \binom{n}{i}(q-1)^i.$$

In the binary case we have

$$|B(\boldsymbol{c}, t)| = \sum_{i=0}^{t} \binom{n}{i}.$$

Proof. First we calculate the number of vectors that are at distance 1 from the codeword $\boldsymbol{c}$. These vectors are the ones that differ from $\boldsymbol{c}$ in exactly one location. There are n possible locations and $q-1$ ways to make an entry different. Thus, the number of vectors that have Hamming distance of 1 from $\boldsymbol{c}$ is $n(q-1)$. Now let's calculate the number of vectors that have Hamming distance m from $\boldsymbol{c}$. There are $\binom{n}{m}$ ways in which we can choose m locations to differ from the values of $\boldsymbol{c}$. For each of these m locations, there are $q-1$ choices for symbols different from the corresponding symbol of $\boldsymbol{c}$. Hence, there are

$$\binom{n}{m}(q-1)^m$$

vectors that have Hamming distance of m from $\boldsymbol{c}$. Including the vector $\boldsymbol{c}$ itself and using the identity $\binom{n}{0} = 1$, we get

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t = |B(\boldsymbol{c}, t)|.$$

□

We may now state the Hamming bound, which is also called the **sphere packing bound**.

Theorem 2.3 (Hamming bound)**.** *Let C be an $(n,\ M,\ d)_q$ code with*

$$d \geq 2t + 1.$$

Then

$$M \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}.$$

Proof. Around each codeword $\boldsymbol{c}$ we place a Hamming sphere of radius t. Since the minimum distance of the code is $d \geq 2t+1$, these spheres do not overlap. The total number of vectors in all of the Hamming spheres can not be greater than q^n. Thus, we get

$$(number\ of\ codewords) \times (number\ of\ elements\ per\ sphere) \leq q^n$$

i.e

$$M \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n$$

□

An $(n, M, d)_q$ code with $d = 2t+1$ that satisfies the Hamming bound with equality is called a **perfect code**. A perfect $t-$error correcting code is one such that the M Hamming spheres of radius t with center at the codewords cover the entire space of $q-$ary $n-$tuples.

a. If we call $V_t(n) = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$, $MV_t(n) \leq q^n$. Then $\log_q(M) + \log_q(V_t(n)) \leq n$, i.e.

$$R \leq 1 - \frac{\log_q(V_t(n))}{n}.$$

Example 2.1. *The following is an interesting example, which is a perfect linear code with $d = 3$, that is, it can correct one error, see [9]. A code defined by a binary parity check matrix with m rows and $2^m - 1$ distinct nonzero columns is a binary Hamming code. If $m = 3$, $n = 2^3 - 1 = 7$ and a parity check matrix is*

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Generally, for each $m \geq 3$ there is a binary Hamming code with parameters $n = 2^m - 1, \quad k = 2^m - 1 - m, \quad d = 3$, which is perfect code because

$$\frac{2^n}{\sum_{i=0}^{1} \binom{n}{i}} = \frac{2^n}{1 + 2^m - 1} = \frac{2^n}{2^m} = 2^{n-m} = 2^k.$$

We observe that the above parity check matrix is in standard form, that is, $H = (P^t | I_{n-k}) = (P^t | I_3)$. Then a generator matrix is given by $G = (I_k | P) = (I_4 | P)$, that is,

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Theorem 2.4. *In a binary Hamming code, every word in $\mathbb{F}_2^n$ is either a codeword or is at a distance 1 from exactly one codeword.*

Proof. There are 2^k codewords, and given a codeword $\boldsymbol{c}$ we know

$$|B(\boldsymbol{c}, t)| = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

Since $q = 2$ and $t = 1$ because $d = 3$, we have

$$|B(\boldsymbol{c}, 1)| = \binom{n}{0} + \binom{n}{1} = 1 + n.$$

Then $2^k |B(\boldsymbol{c}, 1)| = 2^k(1+n) = 2^k + 2^k n = 2^k + 2^k(2^m - 1) = 2^{k+m} = 2^n$. This means that the 2^k codewords $\boldsymbol{c}$ determine disjoint neighborhoods $|B(\boldsymbol{c}, 1)|$ that completely cover $\mathbb{F}_2^n$, as claimed. □

The following theorem describes a very simple way to correct one error in some circumstances.

Theorem 2.5. *Let $C \subseteq \mathbb{F}_2^n$ be a linear code defined by a parity check matrix H. Suppose that a one-bit error is made in transmitting a codeword, the receiver word being $\boldsymbol{z}$. Then the error has occurred in the ith bit of $\boldsymbol{z}$, where i is determined by the fact that $S = H\boldsymbol{z}^t$ is equal to the ith column of H.*

Proof. Suppose that the codeword sent is $\boldsymbol{c}$ and the error is made in the *ith* bit. Then the received word is

$$\boldsymbol{z} = \boldsymbol{c} + \boldsymbol{e}$$

where $\boldsymbol{e} = (\overbrace{0}^{1} \cdots \underbrace{1}_{i} \cdots \overbrace{0}^{n})$. Since $\boldsymbol{c}$ is a codeword, $H\boldsymbol{c}^t = 0$. Then

$$S = H\boldsymbol{z}^t = H\boldsymbol{e}^t = (h_{1i}\ h_{2i} \cdots h_{ni})^t,$$

which is the *ith* column of H. □

Assuming that not more than one-bit error is made in transmitting each codeword, the following procedure can be used:

$Hz^t = 0$?		
YES	NO	
$\boldsymbol{z}$ is a codeword	$H\boldsymbol{z}^t = ith\ column$?	
	YES	NO
	Correct *ith* bit	More than 1 bit error made

Table 2–1

2.1.1 TWO-WEIGHT CODES

Let $C \subset \mathbb{F}_q^n$ be an $[n,\ k,\ d]_q$ code. There exists an integer $r \geq 1$ such that $C \setminus \{\mathbf{0}\} = \cup_{i=1}^{r} C_{w_i}$, where $C_{w_i} = \{\boldsymbol{u} \in C : \omega(\boldsymbol{u}) = w_i\}$, and we call $A_{w_i} = |C_{w_i}|$. For example, let $C \subset \mathbb{F}_4^n$ be given by Table 3–2. Then $C = \{\mathbf{0}\} \cup C_{w_1} \cup C_{w_2}$, where C_{w_1} is given by Table 2–2 and C_{w_2} by Table 2–3.

C_{w_1}			
$(0\ \delta\ \delta\ \delta\ \delta\ 0\ 0)$	$(\delta\ 0\ \delta\ \delta\ 0\ \delta\ 0)$	$(\delta\ \delta\ 0\ \delta\ 0\ 0\ \delta)$	$(\delta\ \delta\ 0\ 0\ \delta\ \delta\ 0)$
$(\delta\ 0\ \delta\ 0\ \delta\ 0\ \delta)$	$(0\ \delta\ \delta\ 0\ 0\ \delta\ \delta)$	$(0\ 0\ 0\ \delta\ \delta\ \delta\ \delta)$	$(0\ 1\ 1\ 1\ 1\ 0\ 0)$
$(0\ \delta^2\ \delta^2\ \delta^2\ \delta^2\ 0\ 0)$	$(1\ 0\ 1\ 1\ 0\ 1\ 0)$	$(\delta^2\ 0\ \delta^2\ \delta^2\ 0\ \delta^2\ 0)$	$(1\ 1\ 0\ 1\ 0\ 0\ 1)$
$(\delta^2\ \delta^2\ 0\ \delta^2\ 0\ 0\ \delta^2)$	$(1\ 1\ 0\ 0\ 1\ 1\ 0)$	$(\delta^2\ \delta^2\ 0\ 0\ \delta^2\ \delta^2\ 0)$	$(1\ 0\ 1\ 0\ 1\ 0\ 1)$
$(\delta^2\ 0\ \delta^2\ 0\ \delta^2\ 0\ \delta^2)$	$(0\ 1\ 1\ 0\ 0\ 1\ 1)$	$(0\ \delta^2\ \delta^2\ 0\ 0\ \delta^2\ \delta^2)$	$(0\ 0\ 0\ 1\ 1\ 1\ 1)$
$(0\ 0\ 0\ \delta^2\ \delta^2\ \delta^2\ \delta^2)$			

Table 2–2

C_{w_2}			
$(\delta\ 1\ \delta^2\ \delta^2\ 1\ \delta\ 0)$	$(\delta\ \delta^2\ 1\ \delta^2\ 1\ 0\ \delta)$	$(\delta\ \delta^2\ 1\ 1\ \delta^2\ \delta\ 0)$	$(\delta\ 1\ \delta^2\ 1\ \delta^2\ 0\ \delta)$
$(0\ \delta^2\ \delta^2\ 1\ 1\ \delta\ \delta)$	$(0\ 1\ 1\ \delta^2\ \delta^2\ \delta\ \delta)$	$(1\ \delta\ \delta^2\ \delta^2\ \delta\ 1\ 0)$	$(\delta^2\ \delta\ 1\ \delta^2\ 0\ 1\ \delta)$
$(\delta^2\ \delta\ 1\ 1\ \delta\ \delta^2\ 0)$	$(\delta^2\ 0\ \delta^2\ 1\ \delta\ 1\ \delta)$	$(1\ \delta\ \delta^2\ 1\ 0\ \delta^2\ \delta)$	$(1\ 0\ 1\ \delta^2\ \delta\ \delta^2\ \delta)$
$(1\ \delta^2\ \delta\ \delta^2\ \delta\ 0\ 1)$	$(\delta^2\ 1\ \delta\ \delta^2\ 0\ \delta\ 1)$	$(\delta^2\ \delta^2\ 0\ 1\ \delta\ \delta\ 1)$	$(\delta^2\ 1\ \delta\ 1\ \delta\ 0\ \delta^2)$
$(1\ \delta^2\ \delta\ 1\ 0\ \delta\ \delta^2)$	$(1\ 1\ 0\ \delta^2\ \delta\ \delta\ \delta^2)$	$(1\ \delta^2\ \delta\ \delta\ \delta^2\ 1\ 0)$	$(\delta^2\ 1\ \delta\ \delta\ 1\ \delta^2\ 0)$
$(\delta^2\ \delta^2\ 0\ \delta\ 1\ 1\ \delta)$	$(\delta^2\ 1\ \delta\ 0\ \delta^2\ 1\ \delta)$	$(1\ \delta^2\ \delta\ 0\ 1\ \delta^2\ \delta)$	$(1\ 1\ 0\ \delta\ \delta^2\ \delta^2\ \delta)$
$(1\ \delta\ \delta^2\ \delta\ \delta^2\ 0\ 1)$	$(\delta^2\ 0\ \delta^2\ \delta\ 1\ \delta\ 1)$	$(\delta^2\ \delta\ 1\ \delta\ 1\ 0\ \delta^2)$	$(\delta^2\ \delta\ 1\ 0\ \delta^2\ \delta\ 1)$
$(1\ \delta\ \delta^2\ 0\ 1\ \delta\ \delta^2)$	$(1\ 0\ 1\ \delta\ \delta^2\ \delta\ \delta^2)$	$(0\ \delta^2\ \delta^2\ \delta\ \delta\ 1\ 1)$	$(\delta\ 1\ \delta^2\ \delta\ 0\ \delta^2\ 1)$
$(\delta\ \delta^2\ 1\ \delta\ 0\ 1\ \delta^2)$	$(\delta\ \delta^2\ 1\ 0\ \delta\ \delta^2\ 1)$	$(\delta\ 1\ \delta^2\ 0\ \delta\ 1\ \delta^2)$	$(0\ 1\ 1\ \delta\ \delta\ \delta^2\ \delta^2)$
$(0\ \delta\ \delta\ \delta^2\ \delta^2\ 1\ 1)$	$(\delta\ 0\ \delta\ \delta^2\ 1\ \delta^2\ 1)$	$(\delta\ \delta\ 0\ \delta^2\ 1\ 1\ \delta^2)$	$(\delta\ \delta\ 0\ 1\ \delta^2\ \delta^2\ 1)$
$(\delta\ 0\ \delta\ 1\ \delta^2\ 1\ \delta^2)$	$(0\ \delta\ \delta\ 1\ 1\ \delta^2\ \delta^2)$		

Table 2–3

From [7], [16] and [22] we have the following definition.

Definition 2.3. *An* $[n,\ k,\ d]_q$ *code* C *such that* $C \setminus \{\mathbf{0}\} = \cup_{i=1}^{r} C_{w_i}$ *is called a* $r-$*weight code. The sequence* $1,\ A_{w_1},\ A_{w_2}, \cdots, A_{w_r}$ *is called the weight distribution of* C*. In addition, if* C *is a two-weight code such that* $w_1 = d$ *and* $w_2 = n$*, the code is called antipodal.*

Table 2–2 and Table 2–3 give us an example of a 2−weight code over $\mathbb{F}_4$. The simplex code $S_k(q) \subset \mathbb{F}_q^n$ of dimension k with $n = \frac{q^k-1}{q-1}$ and $d = q^{k-1}$ is an example of a one-weight code, that is, each nonzero codeword has weight $w_1 = q^{k-1}$, see [7] *Section* 2.5 and *Section* 3.4.

Now, our formulation of linear codes is via the axiomatic approach introduced by Assmus and Mattson in [4]. In this formulation, a linear code C of length n and dimension k over $\mathbb{F}_q$ will be viewed as a pair (V, Λ), in which V is a $k-$dimensional vector space over $\mathbb{F}_q$ and $\Lambda = (\lambda_1 \ \cdots \ \lambda_n)$ is an $n-$tuple of (possibly repeated) linear functionals in the dual space V^* of V. The members of Λ are the coordinate functionals of C and a vector $\boldsymbol{v} \in V$ is encoded as the codeword

$$\boldsymbol{c} = \Lambda(\boldsymbol{v}) = (\lambda_1(\boldsymbol{v}) \ \cdots \ \lambda_n(\boldsymbol{v}))$$

One assumes that Λ satisfies the coding axiom, i.e.,

$$\boldsymbol{v} \to \boldsymbol{c}$$

is one-to-one, that is, $\lambda_i(\boldsymbol{v}) = 0$ for all i, implies $\boldsymbol{v} = 0$. The weight of the codeword $\boldsymbol{c}$ is

$$\omega(\boldsymbol{c}) = |\{i \mid \lambda_i(\boldsymbol{v}) \neq 0\}|,$$

where $\boldsymbol{v} \to \boldsymbol{c}$, see [30] and [52].

The q-ary simplex code of length $n = \frac{q^k-1}{q-1}$ and dimension k, denoted by $S_k(q)$, is then obtained by taking Λ to comprise one nonzero element of each one-dimensional subspace of V^*. One can show that these q-ary simplex codes are constant weight with minimum distance $d = q^{k-1}$ and weight distribution $A_0 = 1, A_d = q^k - 1$.

Definition 2.4. *[52] If $C = (V, \Lambda)$, the $r-$fold replication of C is $(V, r\Lambda)$, where $r\Lambda$ is the multiset in which each member of Λ appears r times (up to scalars).*

The constant weight linear codes over $\mathbb{F}_q$ were characterized as a replication of some $q-$ary simplex code of dimension k-possibly with added 0−coordinates (up to

monomial equivalence). Subsequently, Tonchev et al., [30], characterized all antipodal two-weight linear codes with the minimal number of full weight codewords as a replication of the first-order generalized Reed-Muller codes $R_q(1,k)$ (up to monomial equivalence). One can define the q-ary Reed-Muller codes via extended cyclic codes, via finite geometries, via polynomial codes, or via a recursive construction (see Peterson and Weldon [41] and Hoffman and Pless [28]).

In [52], Ward gives a proof of the Proposition 2.1 using the axiomatic approach in [4]. The characterization of one-weight linear codes was first proved by W. W. Peterson [41] in 1961. There were several more proofs of it published in the 1960s, all independently of each other and of the first [54].

Proposition 2.1. *Let C be a one-weight code over $\mathbb{F}_q$. Then C is equivalent to a replicated simplex code, possibly with added $0-$coordinates.*

Borges, Rifa, and Zinoviev [12] proved the following result.

Theorem 2.6. *Let $C \subset \mathbb{F}_q^n$ be an $[n,\ k+1,\ d]_q$ antipodal linear two-weight code. Then, up to monomial equivalence, C has a generator matrix of the form:*

$$G = \begin{pmatrix} \mathbf{1} & 1 \\ \boldsymbol{M} & \mathbf{0} \end{pmatrix},$$

where $\boldsymbol{M}$ generates a one-weight code E with length $n-1$, dimension k, and distance $d = w_1$, such that for every codeword $\boldsymbol{a}$ in the code spanned by $(\boldsymbol{M}\ \mathbf{0})$, all symbols which occur in $\boldsymbol{a}$ occur exactly $n-d$ times each.

Example 2.2. *The first-order Reed-Muller code $C = \mathbf{R}(1,\ m) \subset \mathbb{F}_2^{2^m}$ is an antipodal linear two-weight code with $w_1 = 2^{m-1}$ and $w_2 = n = 2^m$. In the particular case of $C = \mathbf{R}(1,3)$, a generator matrix for C is given by*

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{1} & 1 \\ G(S_3(2)) & \mathbf{0} \end{pmatrix},$$

where $G(S_3(2))$ is a generator matrix of the binary simplex code of dimension 3.

In [30], Tonchev et al., proved the following result.

Theorem 2.7. *Let C be any antipodal linear two-weight code over $\mathbb{F}_q$ and assume that C contains no linearly independent codewords of full weight. Then C is, up to monomial equivalence, a replication of some first-order generalized Reed-Muller code over $\mathbb{F}_q$.*

From [7], let $\mathbb{F}_q^k$ be the space of the $k-$tuples, a $k-$dimensional vector space over $\mathbb{F}_q$. It makes sense, intuitively, to view the *one-dimensional subspaces* of $\mathbb{F}_q^k$ as *points* and the two-dimensional subspaces as lines. The main reason is that any two points are on precisely one common line (two different one-dimensional subspaces generate a two-dimensional subspace) and it is a familiar axiom in geometry. Observe the shift in dimension: we view one-dimensional subspaces as points (zero-dimensional geometric objects), two-dimensional subspaces as lines (one-dimensional geometric objects) and so forth. Consequently, the geometry derived from $\mathbb{F}_q^k$ is considered to be $(k-1)-$dimensional: $\mathbf{PG}(k-1,q)$, the $(k-1)-$dimensional **projective geometry**. It has $k-1$ types of objects, from points (one-dimensional subspaces) to hyperplanes ((k-1)-dimensional vector subspaces).

Example 2.3. *Considering $k = 3$, we get the projective plane $\mathbf{PG}(2,q)$. For each $\boldsymbol{a} \in \mathbb{F}_q^3$, with $\boldsymbol{a} \neq 0$ and for any $r \in \mathbb{F}_q^*$, we observe that $r\boldsymbol{a}$ represents the same one-dimensional subspace generate by $\boldsymbol{a}$. Then we have $\frac{q^3-1}{q-1} = q^2+q+1$ one-dimensional subspaces in $\mathbb{F}_q^3$, i.e., we obtain q^2+q+1 points in $\mathbf{PG}(2,q)$. Now, given $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{F}_q^3$, with $\boldsymbol{a} \neq s\boldsymbol{b}$ and $s \in F_q^*$, $H = \{r\boldsymbol{a} + s\boldsymbol{b} \mid r,\ s \in \mathbb{F}_q\}$ is*

a two-dimensional subspace in $\mathbb{F}_q^3$. *We observe that,* $r\boldsymbol{a}$ *and* $s\boldsymbol{b}$ *are two different one-dimensional subspaces in* H, *i.e., they are two different points. But* $\boldsymbol{a}+s\boldsymbol{b}$ *is other one-dimensional subspace in* H. *This gives us another* $q-1$ *one-dimensional subspaces in* H. *Then, we have* $(q-1)+1+1=q+1$ *one-dimensional subspaces in* H. *Thus,* $\mathbf{PG}(2,q)$ *has* q^2+q+1 *points (one-dimensional subspaces in* $\mathbb{F}_q^3$*),* $q+1$ *points in each line and* q^2+q+1 *lines (two-dimensional subspaces in* $\mathbb{F}_q^3$*). The smallest projective plane is* $\mathbf{PG}(2,2)$ *(7 points, 7 lines, 3 points in each line), this binary projective plane is also known as the* **Fano plane**. *In general,* $\mathbf{PG}(k-1,q)$ *has* $\frac{q^k-1}{q-1}$ *points and equally many hyperplanes.*

From [16] we have:

Definition 2.5. *A projective* (n,k,h_1,h_2) *set* $\mathbf{O}$ *is a proper, non-empty set of* n *points of the projective space* $\mathbf{PG}(k-1,q)$ *with the property that every hyperplane meets* $\mathbf{O}$ *in* h_1 *points or in* h_2 *points.*

Remark: Given G a $k\times n$ generator matrix of a linear code $C\subset\mathbb{F}_q^n$, if we denote by $A=\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n\}$ the set (it is possible that A is a multiset in $\mathbf{PG}$(k-1,q), i.e., there are two columns generating the same one-dimensional subspace) of the n columns of G, then for any codeword $\boldsymbol{c}\in C$ there is $\boldsymbol{a}\in\mathbb{F}_q^k$ such that

$$\boldsymbol{c}=\boldsymbol{a}G=(\boldsymbol{a}\cdot\boldsymbol{v}_1,\boldsymbol{a}\cdot\boldsymbol{v}_2,\cdots,\boldsymbol{a}\cdot\boldsymbol{v}_n)\,.$$

That is, there is a linear functional $f_{\boldsymbol{a}}\in\ell(\mathbb{F}_q^k,\mathbb{F}_q)$ where $f_{\boldsymbol{a}}(\boldsymbol{x})=\boldsymbol{a}\cdot\boldsymbol{x}=a_1x_1+\cdots+a_kx_k$, i.e.,

$$\boldsymbol{c}=(f_{\boldsymbol{a}}(\boldsymbol{v}_1),\cdots,f_{\boldsymbol{a}}(\boldsymbol{v}_n))$$

and we can define

$$C=\{(f_{\boldsymbol{a}}(\boldsymbol{v}_1)\ \cdots\ f_{\boldsymbol{a}}(\boldsymbol{v}_n))\mid\boldsymbol{a}\in\mathbb{F}_q^k\}. \tag{2.1}$$

Since $dim(C)=k$ and the row rank is equal to the column rank, we get that the columns of G span $\mathbb{F}_q^k$. We know $H=ker(f_{\boldsymbol{a}})=\{\boldsymbol{x}\in\mathbb{F}_q^k\mid f_{\boldsymbol{a}}(\boldsymbol{x})=0\}$ is a $(k-1)-$dimensional subspace in $\mathbb{F}_q^k$, that is, it is a hyperplane in $\mathbf{PG}(k-1,q)$ and

we observe that

$$\omega(\boldsymbol{c}) = |\{i \mid \boldsymbol{v}_i \notin ker(f_{\boldsymbol{a}})\}|,$$

i.e., the Hamming weight of the codeword $\boldsymbol{c} = \boldsymbol{a}G$ is equal to the number of columns in G outside H. Therefore, if $\omega(\boldsymbol{c}) = d$, there are $n - d$ columns of G inside H. If C is a constant weight code, with common weight w_1, then any hyperplane H contains $n - w_1$ columns of the generator matrix G and the other w_1 columns are outside H. From [7] we have

Definition 2.6. *A linear code $[n, k]_q$ is called* **projective** *if it has a generator matrix G whose columns generate different points in $\mathbf{PG}(k-1, q)$, i.e., if no two columns of G are dependent over $\mathbb{F}_q$.*

Example 2.4. *From [7], let q be a prime power and $M_k(q)$ be a matrix with entries in $\mathbb{F}_q$ such that its columns are representatives for the one-dimensional subspaces of $\mathbb{F}_q^k$. This means that there is no $\mathbf{0}$ column, no two columns are multiples of each other, and $M_k(q)$ is maximal with these properties, meaning that, for every nonzero $k-$tuple, some nonzero multiple of it is a column of $M_k(q)$. The number of columns is then $|\mathbf{PG}(k-1, q)| = \frac{q^k-1}{q-1}$. We remember that the* **Hamming code** *$H_k(q)$ is the code with check matrix a $k \times \frac{q^k-1}{q-1}-$matrix $M_k(q)$. Then the $q-$ary simplex code of dimension k is given by $S_k(q) = H_k(q)^{\perp}$, i.e., $S_k(q)$ is a $[\frac{q^k-1}{q-1},\ k,\ q^{k-1}]_q$ projective code.*

From [16] we have

Theorem 2.8. *The code C defined by (2.1) is a projective two-weight $[n,\ k]_q$ code with weights w_1 and w_2 if and only if $\{\prec \boldsymbol{v}_i \succ \ \mid i = 1, \cdots, n\}$ is a projective $(n, k, n - w_1, n - w_2)$ set that spans $\mathbf{PG}(k-1, q)$.*

Proof. Let $\boldsymbol{a}$ be any nonzero vector in $\mathbb{F}_q^k$. If $H = ker(f_{\boldsymbol{a}}) = \{\boldsymbol{x} \in \mathbb{F}_q^k \mid f_{\boldsymbol{a}}(\boldsymbol{x}) = \boldsymbol{a} \cdot \boldsymbol{x} = 0\}$ then $n-|H \cap \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n\}|$ is the weight of the codeword $\boldsymbol{c} = (f_{\boldsymbol{a}}(\boldsymbol{v}_1) \ \cdots \ f_{\boldsymbol{a}}(\boldsymbol{v}_n))$. Observe that, since $dim(C) = k$ and the row rank is equal to column rank, the columns of G span $\mathbb{F}_q^k$, i.e., $\{\prec \boldsymbol{v}_i \succ \ \mid i = 1, \cdots, n\}$ spans $\mathbf{PG}(k-1, q)$. □

2.2 SEPARABLE AND INSEPARABLE GOPPA CODES AND THEIR PARAMETERS: SOME PRELIMINARY RESULTS

We present some properties of the Goppa code $\mathbf{\Gamma}(L, g(x))$, see [38], [39] and [41]. It is defined by the Goppa polynomial $g(x)$ of degree t over the extension field $\mathbb{F}_{q^m}$, where q is a prime power, and an accessory subset L of $\mathbb{F}_{q^m}$. Let

$$g(x) = g_0 + g_1 x + \cdots + g_t x^t \in \mathbb{F}_{q^m}[x] \ \ and \ \ L = \{\alpha_1, \cdots, \alpha_n\} \subset \mathbb{F}_{q^m}$$

be such that $g(\alpha_i) \neq 0$ for all $\alpha_i \in L$. With a vector $\mathbf{c} = (c_1 \ \cdots \ c_n)$ in $\mathbb{F}_q^n$ we associate the function

$$R_c(x) = \sum_{i=1}^{n} \frac{c_i}{x - \alpha_i}$$

where $\frac{1}{x-\alpha_i}$ is the unique polynomial such that $(x - \alpha_i)\frac{1}{x-\alpha_i} \equiv 1 \pmod{g(x)}$. We observe that

$$\frac{1}{x - \alpha_i} = \frac{g(\alpha_i) - g(x)}{x - \alpha_i} g(\alpha_i)^{-1} \pmod{g(x)}$$

Definition 2.7. *The Goppa code $\mathbf{\Gamma}(L, g(x)) \subset \mathbb{F}_q^n$ consists of all vectors $\mathbf{c} = (c_1 \ \cdots \ c_n)$ such that*

$$R_c(x) \equiv 0 \pmod{g(x)}.$$

Before identifying some properties of the Goppa codes, we give a definition of the BCH code and its relation with the Goppa code, see [39].

Definition 2.8. *Let b be a nonnegative integer and $\alpha \in \mathbb{F}_{q^m}$ be a primitive nth root of unity. A BCH code over $\mathbb{F}_q$ of length n and designed distance d, $2 \leq d \leq n$, is a cyclic code defined by the roots $\alpha^b, \alpha^{b+1}, \cdots, \alpha^{b+d-2}$ of the generator polynomial. If $m_i(x)$ denotes the minimal polynomial of α^i over $\mathbb{F}_q$, the generator polynomial of a BCH code has form*

$$g(x) = lcm(m_b(x), m_{b+1}(x), \cdots, m_{b+d-2}(x)) \in \mathbb{F}_{q^m}.$$

Some special cases of the general definition are also important, see [36]. If $b = 1$, the corresponding BCH codes are called narrow-sense BCH codes.

If $n = q^m - 1$, the BCH codes are called primitive.

If $n = q - 1$, a BCH code of length n over $\mathbb{F}_q$ is called a Reed-Solomon code.

Proposition 2.2. *A narrow-sense BCH code is a Goppa code.*

Proof. Let $C = \prec g(x) \succ$ be a narrow-sense BCH code of length n and $\alpha, \alpha^2, \cdots, \alpha^{d-1}$ be the powers of α used to calculate $g(x)$, where $\alpha \in \mathbb{F}_{q^m}$ and $ord(\alpha) = n$. Then $(c_0\ c_1\ \cdots\ c_{n-1}) \in C$ if and only if $c_0 + c_1\alpha_j + c_2\alpha_j^2 + \cdots + c_{n-1}\alpha_j^{n-1} = u(\alpha_j)g(\alpha_j) = 0$ for $1 \leq j \leq d-1$. We write $\alpha_j = \alpha^j$. For each j we write

$$s_j = \sum_{i=0}^{n-1} r_i \alpha_j^i$$

with $\boldsymbol{r} = (r_0\ r_1\ \cdots\ r_{n-1}) \in \mathbb{F}_q^n$. Then $\boldsymbol{r} \in C$ if and only if $s_j = 0$ for each j. We define the polynomial

$$s(x) = s_1 + s_2 x + \cdots + s_{d-1}x^{d-2} = \sum_{j=1}^{d-1} s_j x^{j-1}$$

and obtain that $\boldsymbol{r} \in C$ if and only if $s(\alpha_j) = 0$ for each j. Now,

$$s(x) = \sum_{j=1}^{d-1} \left(\sum_{i=0}^{n-1} r_i \alpha_j^i \right) x^{j-1} = \sum_{i=0}^{n-1} r_i \left(\sum_{j=1}^{d-1} \alpha_j^i\ x^{j-1} \right)$$

and

$$\sum_{j=1}^{d-1} \alpha_j^i\ x^{j-1} = \sum_{j=1}^{d-1} \alpha^{ij}\ x^{j-1} = \alpha^i + \alpha^{2i}x + \cdots + \alpha^{i(d-1)}x^{d-2} = \alpha^i \left(1 + \alpha^i x + \cdots + \alpha^{i(d-2)}x^{d-2}\right).$$

That is,

$$\sum_{j=1}^{d-1} \alpha_j^i\ x^{j-1} = \alpha^i \left[\frac{(\alpha^i x)^{d-1} - 1}{\alpha^i x - 1} \right] = \frac{\alpha^{i(d-1)}x^{d-1} - 1}{x - \alpha^{-i}} \equiv \frac{1}{x - \alpha^{-i}} \pmod{x^{d-1}}.$$

Thus,

$$s(x) \equiv \sum_{i=0}^{n-1} \frac{r_i}{x - \alpha^{-i}} \pmod{x^{d-1}}$$

and therefore $\boldsymbol{r} \in C$ if and only if

$$0 = s(x) \equiv \sum_{i=0}^{n-1} \frac{r_i}{x - \alpha^{-i}} \pmod{x^{d-1}}.$$

We can conclude that a narrow-sense BCH code is a Goppa code with $g(x) = x^{d-1}$ and $L = \{\alpha^{-i} : 0 \leq i \leq n-1\} \subset \mathbb{F}_{q^m}$. □

Theorem 2.9. *The Goppa code $\boldsymbol{\Gamma}(L, g(x))$ of size n and polynomial of degree t is a linear code over $\mathbb{F}_q$ such that $k \geq n - mt$ and $d \geq t + 1$.*

Proof. If we define $\frac{1}{x-\alpha_i} = \frac{g(\alpha_i)-g(x)}{x-\alpha_i} g(\alpha_i)^{-1} \pmod{g(x)}$ we can see $\frac{1}{x-\alpha_i}$ as a polynomial $p_i(x) \in \mathbb{F}_{q^m}[x]$ modulo $g(x)$ of degree $t-1$. We write

$$p_i(x) = a_{i1} + a_{i2}x + \cdots + a_{it}x^{t-1} \pmod{g(x)}.$$

Then

$$0 = \sum_{i=1}^{n} c_i \, p_i(x) = \sum_{i=1}^{n} \left[a_{i1}c_i + a_{i2}xc_i + \cdots + a_{it}x^{t-1}c_i\right].$$

Therefore

$$\sum_{i=1}^{n} a_{ij}c_i = 0 \quad for \quad 1 \leq j \leq t \quad and \quad a_{ij} \in \mathbb{F}_{q^m}$$

We obtain the $n \times t$ matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1t} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nt} \end{pmatrix}.$$

Then we can rewrite the definition of a Goppa code:

$$\boldsymbol{\Gamma}(L, g(x)) = \{\mathbf{c} \in \mathbb{F}_q^n : \ \mathbf{c}A = 0\}.$$

Therefore, $\mathbf{\Gamma}(L, g(x))$ is a linear code over $\mathbb{F}_q$ of length n and can be defined by t linear equations over $\mathbb{F}_{q^m}$, but since

$$a_{ij} = \sum_{k=1}^{m} b_k(ij)\boldsymbol{\beta}_k,$$

where $b_k(ij) \in \mathbb{F}_q$ and $\{\boldsymbol{\beta_1}, \cdots, \boldsymbol{\beta_m}\}$ is a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, $\mathbf{\Gamma}(L, g(x))$ reduces to no more that mt linear equations over $\mathbb{F}_q$ and the dimension of the code must be at least $n - mt$, i.e., $k \geq n - mt$.

Now that we know the code is linear, we can use the fact that for a linear code, the minimum distance is equal to the Hamming minimum weight of the nonzero codewords. We assume $\mathbf{c} \in \mathbf{\Gamma}(L, g(x))$, $\mathbf{c} \neq 0$, $d(\mathbf{c}) = w$. We can write $\mathbf{c} = (c_1\ c_2\ \cdots\ c_w\ 0\ \cdots\ 0)$ doing permutation if necessary. Then

$$R_c(x) = \sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} = \sum_{i=1}^{w} c_i \frac{\prod_{k=1\ i\neq k}^{w}(x - \alpha_k)}{\prod_{k=1}^{w}(x - \alpha_k)}.$$

Since the denominator has no common factor with $g(x)$, because $g(\alpha_i) \neq 0$ for all i, $g(x)$ must be a divisor of the numerator. But the numerator has degree less or equal than $w - 1$, so it follows that

$$w - 1 \geq t$$

and then the minimum distance $d = w \geq t + 1$. Thus,

$$d\left(\mathbf{\Gamma}(L, g(x))\right) \geq t + 1,$$

i.e., $\mathbf{\Gamma}(L, g(x))$ can detect up to t errors.

□

We want the minimum distance of the code to be as large as possible; from Theorem 2.1 a code can correct t errors if $d \geq 2t + 1$. There is a special case where the lower bound on d can be increased. That is the case where $\mathbf{\Gamma}(L, g(x))$ is a binary code, $g(x) \in \mathbb{F}_{2^m}[x]$.

Theorem 2.10. *Let $\mathbf{\Gamma}(L, g(x))$ be a binary Goppa code with a separable polynomial $g(x)$ of degree t. Then*

$$d \geq 2t + 1.$$

Proof. We have $L = \{\alpha_1, \cdots, \alpha_n\} \subset \mathbb{F}_{2^m}$. Let $\mathbf{c} \in \mathbf{\Gamma}(L, g(x))$ with $\mathbf{c} \neq 0$ and $d(\mathbf{c}) = w$. Now, we know from the previous proof that

$$\sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} \equiv 0 \pmod{g(x)} \Leftrightarrow g(x) \mid f(x),$$

where

$$f(x) = \sum_{i=1}^{w} c_i \prod_{k=1\ i\neq k}^{w} (x - \alpha_k) = \sum_{i=1}^{w} \prod_{k=1\ i\neq k}^{w} (x - \alpha_k)$$

because $c_i \in \mathbb{F}_2$ and $c_i \neq 0$. But $\displaystyle\sum_{i=1}^{w} \prod_{k=1\ i\neq k}^{w} (x - \alpha_k)$ is the derivative of the function $\prod_{k=1}^{w}(x - \alpha_k)$ and a binary derivative can only have terms with even exponents, thus

$$f(x) = a_0 + a_2x^2 + a_4x^4 + \cdots + a_{2u}x^{2u}, \ \ with \ \ 2u \leq w - 1.$$

But, for $a \in \mathbb{F}_{2^m}$, we know $a^{2^m} = a$. Then $(a^{2^m})^{\frac{1}{2}} = a^{\frac{1}{2}}$, i.e., $a^{2^{m-1}} = a^{\frac{1}{2}}$ and $a^{2^{m-1}} \in \mathbb{F}_{2^m}$. Thus we can rewrite

$$f(x) = \left(b_0 + b_2x + b_4x^2 + \cdots + b_{2u}x^u\right)^2,$$

where $b_i = a^{\frac{1}{2}}$. So $g(x)$ divides $h^2(x)$ where $h(x) = b_0 + b_2x + bx^2 + \cdots + b_{2u}x^u$, i.e., $h^2(x) = g(x)q(x)$. Since $g(x)$ is separable, $g(x) = (x - d_1)(x - d_2)\cdots(x - d_t)$ with $d_i \neq d_j$ if $i \neq j$. Then $x - d_i \mid h(x)$ for each i. That is, $(x - d_1)(x - d_2)\cdots(x - d_t) \mid h(x)$, i.e., $g(x) \mid h(x)$.

Therefore, $t \leq u$, i.e., $2t \leq 2u \leq w - 1$. Thus

$$d \geq 2t + 1,$$

and a binary Goppa code with a separable polynomial of degree t can correct up to t errors. □

To decode, one needs a **parity check matrix** H of the code. We know $\mathbf{c} \in \mathbf{\Gamma}(L, g(x))$ if and only if

$$\sum_{i=1}^{n} a_{ij} c_i = 0, \quad 1 \leq j \leq t$$

The parity check matrix H satisfies $\mathbf{c}H^t = 0$, thus $H^t = A$.

We are going to determine the factors a_{ij} of the matrix A. We know

$$p_i(x) = (x - \alpha_i)^{-1} = -\frac{g(x) - g(\alpha_i)}{x - \alpha_i} g^{-1}(\alpha_i).$$

Calling $g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_t x^t$ and $h_i = g^{-1}(\alpha_i)$, we obtain

$$p_i(x) = -\frac{g_0 + g_1 x + g_2 x^2 + \cdots + g_t x^t - g_0 - g_1 \alpha_i - g_2 \alpha_i^2 - \cdots - g_t \alpha_i^t}{x - \alpha_i} h_i.$$

That is,

$$p_i(x) = -\frac{g_1(x - \alpha_i) + g_2(x^2 - \alpha_i^2) + \cdots + g_t(x^t - \alpha_i^t)}{x - \alpha_i} h_i,$$

$$p_i(x) = -\left[g_1 + g_2(x - \alpha_i) + \cdots + g_t\left(x^{t-1} + x^{t-2}\alpha_i + \cdots + x^2 \alpha_i^{t-3} + x\alpha_i^{t-2} + \alpha_i^{t-1}\right)\right] h_i,$$

$$p_i(x) = -\left[\left(g_1 + g_2\alpha_i + \cdots + g_t\alpha_i^{t-1}\right) h_i + \left(g_2 + \cdots + g_t\alpha^{t-2}\right) h_i x + \cdots + g_t h_i x^{t-1}\right].$$

Thus,

$$\begin{cases} a_{i1} & = & -\left(g_1 + g_2\alpha_i + \cdots + g_t\alpha_i^{t-1}\right) h_i \\ & & \\ a_{i2} & = & -\left(g_2 + \cdots + g_{t-1}\alpha_i^{t-3} + g_t\alpha_i^{t-2}\right) h_i \\ \vdots & \vdots & \\ a_{i\,t-1} & = & -\left(g_t\alpha_i + g_{t-1}\right) h_i \\ a_{i\,t} & = & -g_t h_i \end{cases}$$

Since $A = H^t$, i.e., $H = A^t$, we obtain that $H = CXY$ where

$$C = -\begin{pmatrix} g_t & g_{t-1} & g_{t-2} & \cdots & g_1 \\ 0 & g_t & g_{t-1} & \cdots & g_2 \\ 0 & 0 & g_t & \cdots & g_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & g_t \end{pmatrix}$$

$$X = \begin{pmatrix} \alpha_1^{t-1} & \alpha_2^{t-1} & \alpha_3^{t-1} & \cdots & \alpha_n^{t-1} \\ \alpha_1^{t-2} & \alpha_2^{t-2} & \alpha_3^{t-2} & \cdots & \alpha_n^{t-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix}$$

$$Y = \begin{pmatrix} h_1 & 0 & 0 & \cdots & 0 \\ 0 & h_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h_n \end{pmatrix}.$$

We observe that $C \in \mathbb{M}_t(\mathbb{F}_{q^m})$, $X \in \mathbb{M}_{t\times n}(\mathbb{F}_{q^m})$ and $Y \in \mathbb{M}_n(\mathbb{F}_{q^m})$, and since C is invertible, another parity check matrix of the code is $H' = XY$.

2.2.1 A SPECIAL CASE

We do not have a specific reference for the Proposition 2.3, which is probably known, but we include it here. We want to find a parity check matrix of a Goppa code when $g(x) = x^l h(x)$, $h(0) \neq 0$, and $\deg(h(x)) = t$. In this case $\deg(g(x)) = l+t$, and a parity check matrix is given by

$$H = \begin{pmatrix} \alpha_1^{t-1} & \alpha_2^{t-1} & \alpha_3^{t-1} & \cdots & \alpha_n^{t-1} \\ \alpha_1^{t-2} & \alpha_2^{t-2} & \alpha_3^{t-2} & \cdots & \alpha_n^{t-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ 1 & 1 & 1 & \cdots & 1 \\ \alpha_1^{-1} & \alpha_2^{-1} & \alpha_3^{-1} & \cdots & \alpha_n^{-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{-l} & \alpha_2^{-l} & \alpha_3^{-l} & \cdots & \alpha_n^{-l} \end{pmatrix} \begin{pmatrix} h^{-1}(\alpha_1) & 0 & 0 & \cdots & 0 \\ 0 & h^{-1}(\alpha_2) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h^{-1}(\alpha_n) \end{pmatrix}.$$

Putting another row of ones in H we obtain the same code, and we can write

$$H = \begin{pmatrix} \alpha_1^{t-1} & \alpha_2^{t-1} & \alpha_3^{t-1} & \cdots & \alpha_n^{t-1} \\ \alpha_1^{t-2} & \alpha_2^{t-2} & \alpha_3^{t-2} & \cdots & \alpha_n^{t-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ \alpha_1^{-1} & \alpha_2^{-1} & \alpha_3^{-1} & \cdots & \alpha_n^{-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{-l} & \alpha_2^{-l} & \alpha_3^{-l} & \cdots & \alpha_n^{-l} \end{pmatrix} \begin{pmatrix} h^{-1}(\alpha_1) & 0 & 0 & \cdots & 0 \\ 0 & h^{-1}(\alpha_2) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h^{-1}(\alpha_n) \end{pmatrix}.$$

If we call $C_1 = \mathbf{\Gamma}(L, h(x))$, $C = \mathbf{\Gamma}(L, x^l h(x))$, and H_1 a parity check matrix of C_1,

$$H = \begin{pmatrix} H_1 \\ \cdots \\ AB \end{pmatrix},$$

where

$$A = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha_1^{-1} & \alpha_2^{-1} & \alpha_3^{-1} & \cdots & \alpha_n^{-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{-l} & \alpha_2^{-l} & \alpha_3^{-l} & \cdots & \alpha_n^{-l} \end{pmatrix} \quad B = \begin{pmatrix} h^{-1}(\alpha_1) & 0 & 0 & \cdots & 0 \\ 0 & h^{-1}(\alpha_2) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h^{-1}(\alpha_n) \end{pmatrix}.$$

Calling $C_0 = \{\mathbf{c} \in \mathbb{F}_q^n : AB\mathbf{c}^t = 0\}$ we obtain $C = C_1 \cap C_0$. Then $d \geq 2t+1$ in the binary case when $h(x)$ is a separable polynomial. We would like to have $d \geq 2(t+l)+1$ in the binary case where $h(x)$ is separable over $\mathbb{F}_{2^m}$, however we have:

Proposition 2.3. *Let $g(x) = x^l h(x)$ be a polynomial over $\mathbb{F}_{2^m}$, where $h(x)$ is separable and such that $h(0) \neq 0$. Then the minimum distance d of $\mathbf{\Gamma}(L, g(x))$ is such that:*

$$d \geq \begin{cases} l + 2t + 1 \; if \; l \; is \; even \\ l + 2t + 2 \; if \; l \; is \; odd. \end{cases}$$

Proof. From *Theorem* 2.10 we have $f(x) = a_0 + a_1 x^2 + a_4 x^4 + \cdots + a_{2u} x^{2u}$ with $2u \leq w - 1$. Since $z^l | f(x)$, $a_0 = 0$ and $f(x) = a_{1_i} x^{l+e_1} + a_{2_i} x^{l+e_2} + \cdots + a_{r_i} x^{l+e_r}$, where $l + e_j$ is an even number. If l is an even number, each e_j is an even number too and

$$f(x) = x^l \left(a_{1_i} x^{e_1} + a_{2_i} x^{e_2} + \cdots + a_{r_i} x^{e_r}\right)$$

Calling $f_1(x) = a_{1_i} x^{e_1} + a_{2_i} x^{e_2} + \cdots + a_{r_i} x^{e_r}$, we can write

$$f_1(x) = \left(b_{1_i} x^{e_1/2} + \cdots b_{r_i} x^{e_r/2}\right)^2 = b(x)^2,$$

where $b_{j_i} = a_{j_i}^2$. Then

$$f(x) = x^l b(x)^2.$$

Since $h(x)|f(x)$ and $h(0) \neq 0$, $h(x)|b(x)^2$ and therefore $h(x)|b(x)$ because $h(x)$ is separable. That is, $t \leq e_r/2$, $2t \leq e_r$ and then $l + 2t \leq l + e_r \leq w - 1$. Thus

$$w \geq l + 2t + 1,$$

if l is even.

If l is an odd number, since all the powers of $f(x)$ are even, we can write

$$f(x) = x^{l+1} \left(a_{i_1} x^{e_1} + \cdots + a_{i_r} x^{e_r} \right),$$

where each e_j is an even number. Then we can write $f(x) = x^{l+1} b(x)^2$. Again, $2t \leq e_r$ and $l + 1 + 2t \leq e_r + l + 1 \leq w - 1$. That is,

$$w \geq l + 2t + 2,$$

if l is an odd number.

□

From *Theorem* 2.1, the binary Goppa code $\mathbf{\Gamma}(L, x^l h(x))$ can correct up $t + \frac{l+1}{2}$ errors.

2.3 BASICS OF QUANTUM ERROR-CORRECTING CODES

In this section, we have some general facts about quantum codes and linear algebra. The principal references are [1, 3, 5, 13, 24, 25, 31, 33, 35, 42, 47], the books [27, 37, 43, 44] and the seminal works [14, 15].

From [44], see *Theorem* 8.10, we have the fact.

Theorem 2.11. *Let $\mathbb{V}$ be a finite dimensional vector space. An operator $T \in Hom(\mathbb{V})$ is diagonalizable if and only if there is a basis for $\mathbb{V}$ that consists entirely of eigenvectors of T; that is, if and only if*

$$\mathbb{V} = \mathbb{V}_{\lambda_1} \oplus \cdots \oplus \mathbb{V}_{\lambda_k},$$

where $\lambda_1, \cdots, \lambda_k$ are the distinct eigenvalues of T.

Given a family $\mathbb{S} \subset Hom(\mathbb{V})$ of diagonalizable operators on a finite-dimensional vector space $\mathbb{V}$, it is necessary that $\mathbb{S}$ be a commuting family to find a basis B of $\mathbb{V}$ such that all the matrices $[T]_B$, $T \in \mathbb{S}$, are diagonal. That follows from the fact that all diagonal matrices commute. From [27], see *Theorem* 6.8, we have:

Theorem 2.12. *Let $\mathbb{S}$ be a commuting family of diagonalizable operators on a finite-dimensional vector space $\mathbb{V}$. There exists an ordered basis for $\mathbb{V}$ such that every operator in $\mathbb{S}$ is represented in that basis by a diagonal matrix.*

Definition 2.9. *Let $\mathbb{V}$ be a vector space over a field $\mathbb{F}$, where $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$. An inner product on $\mathbb{V}$ is a function*

$$(.,.) : \mathbb{V} \times \mathbb{V} \to \mathbb{F}$$

such that:

a. *For all $\boldsymbol{v} \in \mathbb{V}$, $(\boldsymbol{v}, \boldsymbol{v}) \in \mathbb{R}$ and*

$$(\boldsymbol{v}, \boldsymbol{v}) \geq 0 \ \text{ and } \ (\boldsymbol{v}, \boldsymbol{v}) = 0 \leftrightarrow \boldsymbol{v} = 0.$$

In this case, we say that $(.,.)$ is positive defined.

b. *For $\mathbb{F} = \mathbb{C}$,*

$$(\boldsymbol{u}, \boldsymbol{v}) = \overline{(\boldsymbol{v}, \boldsymbol{u})}$$

and for $\mathbb{F} = \mathbb{R}$

$$(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{u}).$$

That is, $(.,.)$ is conjugate symmetric or symmetric.

c. *For all element $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathbb{V}$ and any scalars r, $s \in \mathbb{F}$*

$$(r\boldsymbol{u} + s\boldsymbol{v}, \boldsymbol{w}) = r(\boldsymbol{u}, \boldsymbol{w}) + s(\boldsymbol{v}, \boldsymbol{w})$$

and we say that $(.,.)$ is linear in the first coordinate.

We say that $\mathbb{V}$ *is an inner product space. If in addition* $\mathbb{V}$ *is a* **complete space** *in the metric defined by the norm*

$$||\boldsymbol{v}|| = \sqrt{(\boldsymbol{v}, \boldsymbol{v})},$$

meaning that all Cauchy sequence converge, $\mathbb{V}$ *is called a* **Hilbert space** *and is denoted by* $\mathbb{H}$.

We will take $\mathbb{V}$ as a finite-dimensional space, where the completeness condition always holds and inner product spaces are equivalent to Hilbert spaces. Indeed, we will take $\mathbb{V} = \mathbb{C}^m$ for some integer $m \geq 2$.

Theorem 2.13. *Let* $\mathbb{H}$ *be a finite dimensional inner product space and let* $f \in Hom(\mathbb{H}, \mathbb{F})$. *Then there exists a unique vector* $\boldsymbol{x} \in \mathbb{V}$ *for which* $f(\boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{x})$.

Proof. If f is the zero functional, we can take $\boldsymbol{x} = 0$. Let us assume that $f \neq 0$. Then $\mathbb{K} = ker(f)$ has codimension 1 and since $\mathbb{K}$ is finite dimensional, $\mathbb{H}$ is the orthogonal direct sum of $\mathbb{K}$ and $\mathbb{K}^{\perp}$, i.e.,

$$\mathbb{H} =< \boldsymbol{w} > \odot \mathbb{K}$$

for $\boldsymbol{w} \in \mathbb{K}^{\perp}$. We observe that if $f(\boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{x})$ and $\boldsymbol{x} \in \mathbb{K}$, then in particular $0 = f(\boldsymbol{x}) = (\boldsymbol{x}, \boldsymbol{x})$ and so $\boldsymbol{x} = 0$ because $(.,.)$ is defined positive. Therefore we have $x = \alpha \boldsymbol{w}$ for some $\alpha \in \mathbb{F}$ and $f(\boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{x})$ if and only if

$$f(\boldsymbol{v}) = (\boldsymbol{v}, \alpha \boldsymbol{w}),$$

and since $\boldsymbol{v} \in \mathbb{V}$ has the form $\boldsymbol{v} = \beta \boldsymbol{w} + \boldsymbol{k}$ for $\beta \in \mathbb{F}$ and $\boldsymbol{k} \in \mathbb{K}$, this is equivalent to $f(\beta \boldsymbol{w}) = (\beta \boldsymbol{w}, \alpha \boldsymbol{w})$ or

$$f(\boldsymbol{w}) = \overline{\alpha}(\boldsymbol{w}, \boldsymbol{w}) = \overline{\alpha}||\boldsymbol{w}||^2.$$

Hence, we can take $\alpha = \frac{\overline{f(\boldsymbol{w})}}{||\boldsymbol{w}||^2}$ and

$$\boldsymbol{x} = \frac{\overline{f(\boldsymbol{w})}}{||\boldsymbol{w}||^2}\boldsymbol{w}.$$

□

Theorem 2.14. *Let* $\mathbb{H}$ *and* $\mathbb{H}_1$ *be finite dimensional inner product spaces over* $\mathbb{F}$ *and let* $T \in Hom(\mathbb{H},\ \mathbb{H}_1)$. *Then, there is a unique linear function* $T^* \in Hom(\mathbb{H}_1,\ \mathbb{H})$ *defined by the condition*

$$(T(\boldsymbol{u}), \boldsymbol{v}) = (\boldsymbol{u}, T^*(\boldsymbol{v}))$$

for all $\boldsymbol{u} \in \mathbb{H}$ *and* $\boldsymbol{v} \in \mathbb{H}_1$. T^* *is called the* **adjoint** *of* T.

Proof. For a fixed $\boldsymbol{v} \in \mathbb{H}_1$, consider the function $f_{\boldsymbol{v}} : \mathbb{H} \to \mathbb{F}$ defined by

$$f_{\boldsymbol{v}}(\boldsymbol{u}) = (T(\boldsymbol{u}), \boldsymbol{v}).$$

Then $f_{\boldsymbol{v}} \in Hom(\mathbb{H},\ \mathbb{F})$ and, by Theorem 2.13, there exists a unique vector $\boldsymbol{x} \in \mathbb{H}$ for which

$$f_{\boldsymbol{v}}(\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{x})$$

for all $\boldsymbol{u} \in \mathbb{H}$. Hence, if $T^*(\boldsymbol{v}) = \boldsymbol{x}$ then

$$(T(\boldsymbol{u}), \boldsymbol{v}) = (\boldsymbol{u}, T^*(\boldsymbol{v}))$$

for all $\boldsymbol{u} \in \mathbb{H}$. Now, since

$$\begin{aligned}(\boldsymbol{u}, T^*(r\boldsymbol{v} + s\boldsymbol{w})) &= (T(\boldsymbol{u}), r\boldsymbol{v} + s\boldsymbol{w}) = \overline{r}(T(\boldsymbol{u}), \boldsymbol{v}) + \overline{s}(T(\boldsymbol{u}), \boldsymbol{w}) \\ &= \overline{r}(\boldsymbol{u}, T^*(\boldsymbol{v})) + \overline{s}(\boldsymbol{u}, T^*(\boldsymbol{w})) = (\boldsymbol{u}, rT^*(\boldsymbol{v})) + (\boldsymbol{u}, sT^*(\boldsymbol{w})) \\ &= (\boldsymbol{u}, rT^*(\boldsymbol{v}) + sT^*(\boldsymbol{w}))\end{aligned}$$

for all $\boldsymbol{u} \in \mathbb{H}$, we get that

$$T^*(r\boldsymbol{v} + s\boldsymbol{w}) = rT^*(\boldsymbol{v}) + sT^*(\boldsymbol{w}).$$

Hence $T^* \in Hom(\mathbb{H}_1,\ \mathbb{F})$ and is unique. □

a. We say that $T \in Hom(\mathbb{H})$ is **self-adjoint** (or Hermitian) if $T^* = T$, T is **unitary** (or Orthogonal in the real case) if $TT^* = T^*T = I$, and T is **normal** if $TT^* = T^*T$. Given T a normal operator on $\mathbb{H}$, if λ and μ are distinct eigenvalues of T and $\mathbb{H}_\lambda$ is the eigenspace of λ and $\mathbb{H}_\mu$ the eigenspace of μ, then $\mathbb{H}_\lambda \perp \mathbb{H}_\mu$, because for $\boldsymbol{u} \in \mathbb{H}_\lambda$ and $\boldsymbol{v} \in \mathbb{H}_\mu$,

$$\lambda(\boldsymbol{u}, \boldsymbol{v}) = (T(\boldsymbol{u}), \boldsymbol{v}) = (\boldsymbol{u}, T^*(\boldsymbol{v})) = (\boldsymbol{u}, \overline{\mu}\boldsymbol{v}) = \mu(\boldsymbol{u}, \boldsymbol{v}),$$

and since $\lambda \neq \mu$ we get $(\boldsymbol{u}, \boldsymbol{v}) = 0$.

b. When T is unitary and λ is an eigenvalue, then $|\lambda| = 1$ because

$$(\boldsymbol{u}, \boldsymbol{u}) = (\boldsymbol{u}, T^*T(\boldsymbol{u})) = (T(\boldsymbol{u}), T(\boldsymbol{u})) = \lambda\overline{\lambda}(\boldsymbol{u}, \boldsymbol{u}).$$

That is $\lambda\overline{\lambda} = |\lambda|^2 = 1$.

From [44], see Theorem 10.13, we have:

Theorem 2.15. *Let $\mathbb{H}$ be a finite dimensional inner product space over $\mathbb{C}$. Then an operator T on $\mathbb{H}$ is normal if and only if $\mathbb{H}$ has an orthonormal basis B consisting entirely of eigenvectors of T; that is,*

$$\mathbb{H} = \mathbb{H}_{\lambda_1} \odot \cdots \odot \mathbb{H}_{\lambda_k}$$

where $\{\lambda_1, \cdots, \lambda_k\}$ is the spectrum of T. Put another way, T is normal if and only if it is diagonalizable.

Let $\mathbb{H}$ be a finite dimensional inner product space over $\mathbb{C}$. We can associate any vector $\boldsymbol{v} \in \mathbb{H}$ with an element of $Hom(\mathbb{C},\ \mathbb{H})$ **called ket**, denoted $|\boldsymbol{v}>$, and defined by

$$|\boldsymbol{v}>:\ \mathbb{C} \to \mathbb{H}, \quad \alpha \to \alpha\boldsymbol{v}$$

We regard $|\boldsymbol{v}>$ as the vector itself. The adjoint of a ket, $|\boldsymbol{v}>^*$, is **called bra** and denoted by $<\boldsymbol{v}|$. Then $<\boldsymbol{v}| \in Hom(\mathbb{H},\ \mathbb{C})$ and, by Theorem 2.13, we can write

$$<\boldsymbol{v}| \quad : \quad \mathbb{H} \to \mathbb{C}, \quad \boldsymbol{u} \to (\boldsymbol{u}, \boldsymbol{v}). \tag{2.2}$$

Using this notation, the composition $<\boldsymbol{u}| \circ |\boldsymbol{v}>$ is an element of $Hom(\mathbb{C},\ \mathbb{C})$ for any elements $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathbb{H}$ defined by

$$<\boldsymbol{u}| \circ |\boldsymbol{v}> = <\boldsymbol{u}|\boldsymbol{v}> = (\boldsymbol{v}, \boldsymbol{u}).$$

On the other hand, the composition $|\boldsymbol{u}> \circ <\boldsymbol{v}| = |\boldsymbol{u}><\boldsymbol{v}|$ is an operator on $\mathbb{H}$. For example, if $\mathbb{H}$ is N dimensional over $\mathbb{C}$, $\mathbb{H} \cong \mathbb{C}^N$, an element $\boldsymbol{u} \in \mathbb{H}$ is represented by the $N-$tuple of its component with respect to a specified orthonormal basis:

$$\boldsymbol{u} \to (u_1, \cdots, u_N)^t,$$

and then

$$<\boldsymbol{u}|\boldsymbol{v}> \quad = \quad v_1 u_1^* + \cdots + v_N u_N^* \tag{2.3}$$

is the Hermitian inner product and

$$|u><\boldsymbol{v}| = (u_1, \cdots, u_N)^t (v_1, \cdots, v_N)$$

is an $N \times N$ complex matrix. The matrix $|u><\boldsymbol{v}|$ is known as the outer product of $|\boldsymbol{u}>$ and $|\boldsymbol{v}>$. The outer product of $|\boldsymbol{u}>$ with itself is called density matrix or density operator.

If $\boldsymbol{u} \in \mathbb{H}$ is such that $<\boldsymbol{u}|\boldsymbol{u}> = u_1 u_1^* + \cdots u_N u_N^* = |u_1|^2 + \cdots + |u_N|^2 = 1$, we say that $\boldsymbol{u}$ is a **normalized vector**. Given a Hilbert space $\mathbb{H}$, any physical state can be represented by a normalized vector $|\boldsymbol{u}>$, which is unique up to a phase factor; that is, any two vectors $|\boldsymbol{u}>$ and $|\boldsymbol{v}>$ such that $|\boldsymbol{v}> = r|\boldsymbol{u}>$, with $r \in \mathbb{C}$ and $|r| = 1$, represent the same state. We say that $\mathbb{H}$ is the state space of the system.

Definition 2.10. *Let $\mathbb{H}$ be a space state and U a mapping that takes as input state $|\boldsymbol{u}>$ and output a different state $U|\boldsymbol{u}>$. Then U is a unitary operator which* **is unique up to a phase factor**.

Unitary evolution implies reversibility; that is, we can determine the input state of an evolution given the output state and knowledge of the evolution because unitary operators have an inverse. And since $(U\boldsymbol{u}, U\boldsymbol{v}) = (\boldsymbol{u}, U^*U\boldsymbol{v}) = (\boldsymbol{u}, \boldsymbol{v})$, the unitary evolution preserves the unit norm constrain.

2.3.1 BINARY STABILIZER CODES

It is known that any channel comes with an underlying alphabet, where the letters of the alphabet are the smallest unit of information that can be sent across the channel. In classical error-correcting codes, the alphabet can be the finite field $\mathbb{F}_q$. In the quantum scenario, the analogous to $\mathbb{F}_q$ is a finite-dimensional Hilbert space $\mathbb{H}$, we take $\mathbb{H} = \mathbb{C}^q$ or in particular $\mathbb{H} = \mathbb{C}^2$ in the binary case, see [34], and [45].

Definition 2.11. *The basic unit of quantum information is the* **quantum bit**, *coined as* **qubit** *by Schumacher [45], and its state space is the two state space $\mathbb{C}^2$. The basis states are denoted in the Dirac notation by $|\mathbf{0}>$ and $|\mathbf{1}>$, where $|\mathbf{0}>$ is the column vector $\binom{1}{0} \in \mathbb{C}^2$ and $|\mathbf{1}>$ is the column $\binom{0}{1} \in \mathbb{C}^2$. With this notation a qubit is any system whose state vector $|\boldsymbol{u}>$ can be written as $|\boldsymbol{u}>= u_1|\mathbf{0}> + u_2|\mathbf{1}>$ where $u_1, u_2 \in \mathbb{C}$ and $|u_1|^2 + |u_2|^2 = 1$.*

An important difference with respect to bits is that qubits can be in superposition, any linear combination, of the basis states subject to the normalization requisite. The state space for n qubits is the tensor product of n copies of the state space $\mathbb{C}^2$, i.e., is the Hilbert space $\mathbb{C}^{2^n} \cong \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$. An orthonormal basis for the state space of n qubits is given by $B = \{|a_1 a_2 \cdots a_n >:= |\boldsymbol{a}> \mid \boldsymbol{a} \in \mathbb{F}_2^n\}$; that

is, a general quantum state for n qubits is given by

$$\sum_{\boldsymbol{a}\in\mathbb{F}_2^n} \lambda_{\boldsymbol{a}}|\boldsymbol{a}> \quad and \quad \sum_{\boldsymbol{a}\in\mathbb{F}_2^n} |\lambda_{\boldsymbol{a}}|^2 = 1,$$

where $|a_1a_2\cdots a_n>=|a_1>\otimes\cdots\otimes|a_n>$ is the tensor product of the individual states $|a_i>$. We remember that the tensor product is an associative operation and

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = (a_1b_1\ a_1b_2\ a_2b_1\ a_2b_2)^t.$$

Example 2.5. *For $n=3$, the state space for three qubits is $\mathbb{C}^{2^3}$ and an orthonormal basis is given by*

$$\begin{aligned} \mathbf{B} &= \{|000>,\ |100>,\ |010>,\ |001>,\ |110>,\ |101>,\ |011>,\ |111>\} \\ &= \{|a_1a_2a_3> | \boldsymbol{a}=(a_1\ a_2\ a_3)\in\mathbb{F}_2^3\}, \end{aligned}$$

where

$$\begin{aligned} |000> &= \begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix} = (\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix})\otimes\begin{pmatrix}1\\0\end{pmatrix} = (1000)^t\otimes\begin{pmatrix}1\\0\end{pmatrix} \\ &= (10000000)^t, \\ |100> &= \begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix} = (\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix})\otimes\begin{pmatrix}1\\0\end{pmatrix} = (0010)^t\otimes\begin{pmatrix}1\\0\end{pmatrix} \\ &= (00001000)^t, \end{aligned}$$

and so on.

Example 2.6a. *If $|\boldsymbol{u}>=|\mathbf{0}>$, the quantum state $|\mathbf{1}>$ is an evolution of $|\mathbf{0}>$ because $U=\sigma_x$ is such that $\sigma_x|\mathbf{0}>=|\mathbf{1}>$, where*

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

is **the bit error Puali Matrix or the NOT gate**, *the unitary operator which flips the basis states $|\mathbf{0}>$ and $|\mathbf{1}>$.*

Indeed,

$$\sigma_x|\mathbf{0}>=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix}=\begin{pmatrix}0\\ 1\end{pmatrix}=|\mathbf{1}>$$

and

$$\sigma_x|\mathbf{1}>=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix}=\begin{pmatrix}1\\ 0\end{pmatrix}=|\mathbf{0}>.$$

Let

$$\sigma_z=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$$

be **the phase error Puali matrix**. *Then*

$$\sigma_z|\mathbf{0}>=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix}=\begin{pmatrix}1\\ 0\end{pmatrix}=|\mathbf{0}>$$

and

$$\sigma_z|\mathbf{1}>=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix}=-\begin{pmatrix}0\\ 1\end{pmatrix}=-|\mathbf{1}>.$$

Let

$$\sigma_y=\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}=i\sigma_x\sigma_z$$

be **the bit and phase error Puali matrix**. *Then*

$$\sigma_y|\mathbf{0}>=i|\mathbf{1}>\quad\sigma_y|\mathbf{1}>=-i|\mathbf{0}>.$$

In this example, an important case is when the input quantum state is $|\mathbf{0}>$ *or* $|\mathbf{1}>$ *and the output quantum state is* $|+>=\frac{1}{\sqrt{2}}(|\mathbf{0}>+|\mathbf{1}>)$ *or* $|->=\frac{1}{\sqrt{2}}(|\mathbf{0}>-|\mathbf{1}>)$ *respectively. In this case we can take the unitary transformation given by the unitary matrix* $U=\frac{1}{\sqrt{2}}H_2$ *where*

$$H_2=\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

is the second order Hadamard matrix, because

$$U|\mathbf{0}>=|+> \quad and \quad U|\mathbf{1}>=|->.$$

Also observe that

$$\sigma_z|+>=|-> \quad \sigma_z|->=|+>$$

and

$$\sigma_x|+>=|+> \quad \sigma_x|->=-|->.$$

That is, a phase error σ_z in the basis $\{|\mathbf{0}>, |\mathbf{1}>\}$, corresponds to a bit error in the rotated basis $\{|+>=\frac{1}{\sqrt{2}}(|\mathbf{0}>+|\mathbf{1}>), |->=\frac{1}{\sqrt{2}}(|\mathbf{0}>-|\mathbf{1}>)\}$. See [15] and [34].

Table 2–4 summarizes the action of the Pauli matrices on the state basis for a single qubit.

$\|\boldsymbol{u}>$	σ_x	σ_y	σ_z
$\|\mathbf{0}>$	$\|\mathbf{1}>$	$i\|\mathbf{1}>$	$\|\mathbf{0}>$
$\|\mathbf{1}>$	$\|\mathbf{0}>$	$-i\|\mathbf{0}>$	$-\|\mathbf{1}>$

Table 2–4

From Table 2–4 we have $\sigma_y = i\sigma_x\sigma_z$. Then, in binary quantum context, we have two types of error: σ_x causes **bit error** *and σ_z causes* **phase error**.

b. *If the initial quantum state is given by $|\boldsymbol{v}>=|0101>\in\mathbb{C}^{2^4}$, the state $-|1111>$ is an evolution of $|\boldsymbol{v}>$ because*

$$-|1111>=\sigma_x\otimes I\otimes\sigma_x\otimes\sigma_z|0101>=U|0101>.$$

Observe that we may write $U=-1U_1$ where $U_1=\sigma_x\otimes\sigma_z\otimes\sigma_x\otimes I$ and again

$$U_1|0101>=-|1111>.$$

Remember that unitary evolution **is unique up to a phase factor**. *In the first case we can write the unitary transformation as*

$$\hat{U} := i^0 U_{\boldsymbol{a}} V_{\boldsymbol{b}},$$

where

$$\mathbf{a} = (1\ 0\ 1\ 0)$$

and

$$\boldsymbol{b} = (0\ 0\ 0\ 1)$$

that is, we have bit error, σ_x, at the first and third positions of the input state and phase error, σ_z, at the fourth position. If no error at a position, the identity matrix, we put zero at the respective place. Then

$$U_{\boldsymbol{a}}|\boldsymbol{v}>=|\boldsymbol{v}+\boldsymbol{a}>$$

and

$$V_{\boldsymbol{b}}|\boldsymbol{v}>=(-1)^{\boldsymbol{b}\cdot\boldsymbol{v}}|\boldsymbol{v}>,$$

where $\boldsymbol{b}\cdot\boldsymbol{v}$ means the usual inner product between elements of $\mathbb{F}_2^4$ and $\hat{U}|\boldsymbol{v}>=(-1)^{\boldsymbol{b}\cdot\boldsymbol{v}}|\boldsymbol{v}+\boldsymbol{a}>$, see [14] and [34].

The loss of coherence, **called decoherence**, caused by vibrations, temperature fluctuations, electromagnetic waves, and other interactions with the outside environment, destroys the quantum property of the superposition. For this reason, decoherence represents a challenge for the practical realization of quantum computers, since such machines require the coherence of states to be preserved and that decoherence is managed to perform quantum computation. The preservation of coherence and mitigation of decoherence effects are thus related to the concept of quantum error correction, see [15].

Definition 2.12. *A quantum error-correcting code, Q, with rate $\frac{k}{n}$ is a unitary mapping of $\mathbb{C}^{2^k}$ onto a 2^k-dimensional subspace of $\mathbb{C}^{2^n}$. The subspace itself will be called the quantum error-correcting code.*

From Table 2–4, we can see that a bit error in an individual qubit corresponds to applying σ_x to that qubit, and a phase error to the matrix σ_z. In view of the linearity of quantum mechanics, if we can correct errors E and E_1, we can correct any linear combination of them, $rE + sE_1$, where $r, s \in \mathbb{C}$. That is, we only need to consider whether the code can correct a **basis of errors**. For example, on a two dimensional Hilbert space, a qubit, the most commonly used basis of error operators consists of the four matrices

$$\mathbf{B} \quad = \quad \{I_2,\ \sigma_x,\ \sigma_y,\ \sigma_z\}. \tag{2.4}$$

If we make the assumption that noise on each qubit is independent, it is possible to decompose an error on the system into a tensor product of n single qubit errors. The set $\mathbf{B}$ is an orthonormal basis of linear operators, where the inner product of operators E and E_1 is given by $(E, E_1) = \frac{1}{2}Tr(\overline{E}^t E_1)$, see [14] and [25]. The set describing the possible errors in n qubits is the tensor product of the elements in $\mathbf{B}$, i.e., $\xi_n = \{i^r w_1 \otimes \cdots \otimes w_n | w_i \in \mathbf{B},\ \ r \in \mathbb{Z}\}$, which is a subgroup of the unitary group $U(2^n)$. The **weight of an operator** of this form is the number of qubits on which it differs from the identity. For example, if $E = \sigma_x \otimes I_2 \otimes \sigma_z \otimes \sigma_x$, the weight of $E \in \xi_4$, is three. In general, since the quantum evolution is unitary and unique up to a phase factor, we can write each element of ξ_n uniquely in the form (see [14])

$$E \quad = \quad i^r U_{\boldsymbol{a}} V_{\boldsymbol{b}}, \tag{2.5}$$

where $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_2^n$, $U_{\boldsymbol{a}}, V_{\boldsymbol{b}} \in \xi_n$ and defined by

$$U_{\boldsymbol{a}}|\boldsymbol{u}> = |\boldsymbol{u}+\boldsymbol{a}> \tag{2.6}$$

$$V_{\boldsymbol{b}}|\boldsymbol{u}> = (-1)^{\boldsymbol{u}\cdot\boldsymbol{b}}|\boldsymbol{u}> \tag{2.7}$$

$$E|\boldsymbol{u}> = (-1)^{\boldsymbol{u}\cdot\boldsymbol{b}}|\boldsymbol{u}+\boldsymbol{a}>, \tag{2.8}$$

where $\boldsymbol{u}\cdot\boldsymbol{b}$ means the Euclidean inner product. That is, there are bit errors, σ_x, in the qubits for which $a_j = 1$ and phase errors, σ_z, in the qubits for which $b_j = 1$. By uniqueness, up to a phase factor, of writing each element of ξ_n we associate to the unitary operator $U_{\boldsymbol{a}}V_{\boldsymbol{b}}$ the element $(\boldsymbol{a},\boldsymbol{b}) \in \mathbb{F}_2^{2n}$ to which we associate the element $\varphi(\boldsymbol{a},\boldsymbol{b}) = \boldsymbol{a}+\delta\boldsymbol{b} = (a_1+\delta b_1, \cdots, a_n+\delta b_n) \in \mathbb{F}_4^n$ where φ is a bijective map, $\delta \in \mathbb{F}_4 \setminus \mathbb{F}_2$ and $\delta^2 = \delta + 1$, see Definition 3.1.

Let $E = i^r U_{\mathbf{a}}V_{\mathbf{b}}$ and $E_1 = i^{r_1}U_{\mathbf{c}}V_{\mathbf{d}}$ be given by (2.5). We get that

$$EE_1|\boldsymbol{v}> = Ei^{r_1}(-1)^{\mathbf{d}\cdot\mathbf{v}}|\mathbf{v}+\mathbf{c}> = i^{r+r_1}(-1)^{\mathbf{d}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c}}|\mathbf{v}+\mathbf{c}+\mathbf{a}>$$

and

$$E_1E|\boldsymbol{v}> = i^{r_1+r}(-1)^{\mathbf{b}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{a}}|\mathbf{v}+\mathbf{a}+\mathbf{c}> .$$

Then $EE_1 = E_1E$ if and only if

$$\mathbf{d}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c} = \mathbf{b}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{v}+\mathbf{a}\cdot\mathbf{d}$$

for any $|\mathbf{v}>$; that is,

$$\mathbf{a}\cdot\mathbf{d} = \mathbf{b}\cdot\mathbf{c}.$$

Since we are in characteristic two, $EE_1 = E_1E$ if and only if $\mathbf{a}\cdot\mathbf{d}+\mathbf{b}\cdot\mathbf{c} = 0$, but writing $\boldsymbol{a}+\delta\boldsymbol{b}$ from E and $\boldsymbol{c}+\delta\boldsymbol{d}$ from E_1, then $EE_1 = E_1E$ if and only if $\boldsymbol{a}+\delta\boldsymbol{b}$ and $\boldsymbol{c}+\delta\boldsymbol{d}$ are orthogonal with respect to Trace Hermitian inner product, defined by,

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_{TH}(\boldsymbol{c}+\delta\boldsymbol{d}) = Tr[(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_H(\boldsymbol{c}+\delta\boldsymbol{d})] = \boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}$$

see Equation (3.14) which is a particular case of the Equation (3.5). *Thus a subgroup* $\mathbb{S}$ *of* ξ_n *is commutative if and only if its image* $\overline{\mathbb{S}}$ *through* φ *in* $\mathbb{F}_4^n$ *is self-orthogonal with respect to Trace Hermitian inner product.* From Theorem 2.12, if $\overline{\mathbb{S}} \subset \overline{\mathbb{S}}^{\perp_{TH}}$, the elements of $\mathbb{S}$ can be simultaneously diagonalizable and, from Theorem 2.15, this induces a decomposition of $\mathbb{C}^{2^n}$ into orthogonal eigenspaces. From [3, 25, 31] we have:

Definition 2.13. *The stabilizer* $\mathbb{S}$ *is some abelian subgroup of* ξ_n *and the coding space* $Q \subset \mathbb{C}^{2^n}$ *is the space of vectors fixed by* $\mathbb{S}$*. That is*

$$Q = \bigcap_{E \in \mathbb{S}} \{|\boldsymbol{v}> \in \mathbb{C}^{2^n} | E|\boldsymbol{v}> = |\boldsymbol{v}>\}.$$

Q *is called a stabilizer code and it is the space with all eigenvalues* $+1$*.*

From Definition 2.13, $\mathbb{S}$ is the set of errors, E, that have no effect on the encoded state, $\mathbb{S}$ is the analog of the zero subgroup in the classical coding case. When $\overline{\mathbb{S}}$ is self-orthogonal, we get that for all $E \in \mathbb{S}$ and all $E_1 \in \varphi^{-1}(\overline{\mathbb{S}}^{\perp_{TH}}) = \mathbb{S}^{\perp_{TH}}$, $EE_1 = E_1E$. **Then** $\mathbb{S}^{\perp_{TH}} \backslash \mathbb{S}$ **is the set of the undetectable errors.** Observe that Q is invariant under the elements of $\mathbb{S}^{\perp_{TH}}$. On the other hand, the errors which fail to commute with some element of $\mathbb{S}$ move codewords into an orthogonal subspace to the code, so can be detected by Q, see [5].

Definition 2.14. *Let* $Q \subset \mathbb{C}^{2^n}$ *be a quantum code and* $E, E_1 \in \xi_n$ *be two errors. The code can correct them if and only if* $E(|\boldsymbol{u}>) \neq E_1(|\boldsymbol{v}>)$ *for all* $|\boldsymbol{u}>, |\boldsymbol{v}> \in Q$*; i.e,* $E^{-1}E_1(|\boldsymbol{v}>) \notin Q$*. That is,*

$$0 =< \boldsymbol{u}|E^{-1}E_1|\boldsymbol{v}> =< \boldsymbol{u}|E^*E_1|\boldsymbol{v}> = (E_1|\boldsymbol{v}>, E|\boldsymbol{u}>).$$

A quantum code Q has **minimum distance** d if and only if it can detect all error in ξ_n of weight less than d, but can not detect some error of weight d. When Q is a subspace of $\mathbb{C}^{2^n}$ with dimension 2^k and minimum distance d we say that Q is an $[[n,\ k,\ d]]_2$ quantum code, and **it is pure to** t if and only if its stabilizer subgroup

$\mathbb{S}$ does not contain nonscalar matrices of weight less than t. When $t = d$, we say that Q is pure, see [31].

The following theorem from [14] is useful to us when we apply our construction to stabilizer codes.

Theorem 2.16. *Suppose C is an additive self-orthogonal subcode of $\mathbb{F}_4^n$ with respect to the Trace Hermitian inner product, containing 2^{n-k} vectors, such that there are no vectors of weight less that d in $C^{\perp_{TH}} \setminus C$. Then, any eigenspace of $\varphi^{-1}(C)$ is a quantum error correcting code with parameters $[[n,\ k,\ d]]_2$.*

2.3.2 NONBINARY STABILIZER CODES

Let $q = p^m$ be a power of a prime p and let $\mathbb{F}_q$ be the finite field with q elements. The trace function from $\mathbb{F}_q$ to $\mathbb{F}_p$, is defined by

$$tr(\alpha) = \sum_{i=0}^{m-1} \alpha^{q^i}.$$

Definition 2.15. *A $q-$ary quantum code Q of length n is a q^k dimensional subspace of $\mathbb{C}^{q^n}$.*

The Hilbert space $\mathbb{C}^{q^n}$ is identified with the $n-$fold tensor product of the Hilbert space $\mathbb{C}^q$, and $\mathbb{C}^q$ is thought of as the state space of a $q-$ary system in the same way as the values 0 and 1 can be thought of as the possible states of a bit in a bit string, see [3].

Given two complex or real $n \times m$ matrices A and B, the Frobenius inner product is given by

$$(A, B)_F \quad = \quad \sum_{i,j} b_{ij}\overline{a_{i,j}} = Tr\left(\overline{A}^t B\right). \tag{2.9}$$

When A and B are $n \times 1$ column vectors, the Frobenius inner product corresponds to the Hermitian inner product, see (2.3).

In order to select an appropriate error model in a $q-$ary quantum stabilizer code, we take $\mathbb{H}$ to be a complex Hilbert space of dimension m. We want to find a basis of the complex vector space $Hom(\mathbb{H},\mathbb{H})$ representing a discrete set of errors. Such a basis contains m^2 complex linear independent operators, $\mathbf{B}=\{E_1,\cdots,E_{m^2}\}$, with the following properties:

a. It is a set of unitary operators and $I\in\mathbf{B}$.

b. Taking the matrix representation and the Frobenius inner product, $\mathbf{B}$ is a set of orthonormal unitary operators, that is, $(E_i,E_j)_F=m\delta_{i,j}$. Taking $E_i=I$, we get $(I,E_j)_F=Tr(E_j)=0$ for all $E_j\neq I\in\mathbf{B}$.

c. Since the set of unitary operators is a group under composition, we take $\mathbf{B}$ such that $E_iE_j=w_{ij}E_{i\star j}$ for some operation $\star$ on the set of indices. Observe that

$$I=\overline{(E_iE_j)}^t(E_iE_j)=\overline{w_{ij}}w_{ij}\overline{E_{i\star j}}^tE_{i\star j}=|w_{ij}|^2I,$$

then $|w_{ij}|^2=1$.

In this case **B is called a nice error basis**. Now, since each E_i is a unitary operator, $|det(E_i)|=1$. If in addition, we take each E_i such that $det(E_i)=1$, the basis **B is called a very nice error basis**. For example, taking $\mathbb{H}=\mathbb{C}^2$, (2.4) is a nice error basis for $Hom(\mathbb{C}^2,\mathbb{C}^2)$ and

$$\mathbf{B} \;=\; \{I_2,\ i\sigma_x,\ i\sigma_y,\ i\sigma_z\} \tag{2.10}$$

is a very nice error basis, see [33].

Taking $\mathbb{H}=\mathbb{C}^q$, for $a,b\in\mathbb{F}_q$, we define the unitary operators U_a, V_b and E on $\mathbb{C}^q$ by

$$U_a|x> \;=\; |x+a>, \tag{2.11}$$

$$V_b|x> \;=\; w^{tr(bx)}|x>, \tag{2.12}$$

$$E_{ab}|x> \;=\; =U_aV_b|x>=w^{tr(bx)}|x+a>, \tag{2.13}$$

where $w = exp(2\pi i/p)$ is a primitive pth root of unity and tr denotes the trace operation from the extension $\mathbb{F}_q$ to its prime field $\mathbb{F}_p$. Then

$$\mathbf{B}_1(q) = \{E_{ab} : a,\ b \in \mathbb{F}_q\}$$

is a nice error basis on $\mathbb{C}^q$, compare with (2.4). For $\boldsymbol{a} = (a_1\ \cdots\ a_n) \in \mathbb{F}_q^n$, we write $U_{\boldsymbol{a}} = U_{a_1} \otimes \cdots \otimes U_{a_n}$ for the tensor products of n error operators. The set

$$\mathbf{B}_n(q) = \{E_{\boldsymbol{ab}} = U_{\boldsymbol{a}} V_{\boldsymbol{b}} : \boldsymbol{a},\ \boldsymbol{b} \in \mathbb{F}_q^n\} \tag{2.14}$$

is a nice error basis on the Hilbert space $\mathbb{C}^{q^n}$, see [31, 33].

Since

$$E_{\boldsymbol{ab}} E_{\boldsymbol{cd}} = w^{tr(\boldsymbol{b}\cdot\boldsymbol{c})} U_{\boldsymbol{a}+\boldsymbol{c}} V_{\boldsymbol{b}+\boldsymbol{d}} = w^{tr(\boldsymbol{b}\cdot\boldsymbol{c})} E_{\boldsymbol{a}+\boldsymbol{c}\ \boldsymbol{b}+\boldsymbol{d}}, \tag{2.15}$$

the set

$$\xi_n = \{w^r E_{\boldsymbol{ab}} : \boldsymbol{a},\ \boldsymbol{b} \in \mathbb{F}_q^n,\ r \in \mathbb{F}_p\} \tag{2.16}$$

is a finite group of order pq^{2n}, called **the error group** associated with the nice error basis $\mathbf{B}_n(q)$, see [3] and [31].

Given an abelian subgroup $\mathbb{S}$ of the error group, a $q-$ary stabilizer code Q is defined as

$$Q = \bigcap_{E_{\boldsymbol{ab}} \in \mathbb{S}} \{|\boldsymbol{v}> \in \mathbb{C}^{q^n} | E_{\boldsymbol{ab}} |\boldsymbol{v}> = |\boldsymbol{v}>\}.$$

An important property of a stabilizer code Q is that it contains all joint eigenvectors of $\mathbb{S}$ with eigenvalue 1. Now, we take the centralizer of $\mathbb{S}$ in ξ_n

$$\{E \in \xi_n : EE_1 = E_1 E\ \ \forall E_1 \in \mathbb{S}\},$$

and let $SZ(\xi_n)$ be the group generated by the abelian subgroup $\mathbb{S}$ and the center of ξ_n, $Z(\xi_n)$. From [31], we have the following fact

Proposition 2.4. *Suppose that $\mathbb{S}$ is the stabilizer group of a stabilizer code Q of dimension $\dim Q > 1$. An error E in ξ_n is detectable by the quantum code Q if and*

only if either E is an element of $SZ(\xi_n)$ or E does not belong to the centralizer of $\mathbb{S}$ in ξ_n.

In particular, a stabilizer code Q with stabilizer $\mathbb{S}$, can detect all errors in ξ_n that are scalar multiples of elements in $\mathbb{S}$ or that do not commute with some element of $\mathbb{S}$. From Equation (2.15) we have

Proposition 2.5. *Two elements $E = w^r E_{\boldsymbol{ab}}$ and $E_1 = w^{r_1} E_{\boldsymbol{cd}}$ of the error group ξ_n satisfy the relation*

$$E_1 E = w^{tr(\boldsymbol{a}\cdot\boldsymbol{d}-\boldsymbol{b}\cdot\boldsymbol{c})} E E_1.$$

In particular, the elements E and E_1 commute if and only if $tr(\boldsymbol{a}\cdot\boldsymbol{d}-\boldsymbol{b}\cdot\boldsymbol{c}) = 0$.

When $q = p$, we get $tr(\boldsymbol{a}\cdot\boldsymbol{d}-\boldsymbol{b}\cdot\boldsymbol{c}) = \boldsymbol{a}\cdot\boldsymbol{d}-\boldsymbol{b}\cdot\boldsymbol{c}$ and if $q = p = 2$ it corresponds to Equation (3.14). The weight of an element $E_{\boldsymbol{ab}} \in \xi_n$ is the number of nonidentity tensor components and the weight of a scalar multiple of the identity matrix is zero.

Definition 2.16. *A quantum code Q has minimum distance d if and only if it can detect all errors in ξ_n of weight less than d, but can not detect some error of weight d. A $q-$ary stabilizer code of length n, dimension q^k and minimum distance d is denoted $[[n,\ k,\ d]]_q$.*

In order to get a relation between additive or linear codes in $\mathbb{F}_{q^2}^n$ with stabilizer codes, we define a bijective map φ that takes an element $(\boldsymbol{a},\boldsymbol{b}) \in \mathbb{F}_q^{2n}$ to a vector $\boldsymbol{a}\alpha + \boldsymbol{b}\alpha^q \in \mathbb{F}_{q^2}^n$, where $\{\alpha, \alpha^q\}$ is a normal basis of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$, see Definition 3.2. For $\boldsymbol{x} = \boldsymbol{a}\alpha + \boldsymbol{b}\alpha^q$ and $\boldsymbol{y} = \boldsymbol{c}\alpha + \boldsymbol{d}\alpha^q$ in $\mathbb{F}_{q^2}^n$, we define an alternating form by

$$\boldsymbol{x} *_a \boldsymbol{y} \quad = \quad tr\left(\frac{\boldsymbol{x}\cdot\boldsymbol{y}^q - \boldsymbol{x}^q\cdot\boldsymbol{y}}{\alpha^{2q}-\alpha^2}\right) = tr\left(\boldsymbol{b}\cdot\boldsymbol{c} - \boldsymbol{a}\cdot\boldsymbol{d}\right). \tag{2.17}$$

It is alternating form in the sense that $\boldsymbol{x} *_a \boldsymbol{x} = 0$. This alternating form is bilinear over $\mathbb{F}_p$, i.e., it is a sympletic inner product. From [31], Theorem 15, we have the following fact:

Theorem 2.17. *An* $[[n,\ k,\ d]]_q$ *stabilizer code exists if and only if there exists an additive subcode* D *of* $\mathbb{F}_{q^2}^n$ *of cardinality* $q^n/q^k = q^{n-k}$ *such that* D *is* $*_a-$*self-orthogonal and the weight of* $D^{\perp_a} \setminus D$ *is* d *if* $k > 0$*, and the weight of* $D^{\perp_a}$ *is* d *if* $k = 0$.

CHAPTER 3
GO-UP CONSTRUCTION AND APPLICATIONS

In this chapter, we present a new code construction technique that permits us to get additive codes, two-weight and three-weight codes. **Section 3.1** introduces this technique that we call Go-Up construction. We present an equivalence between a Frobenius invariant linear code over $\mathbb{F}_{q^m}$ and a code obtained from our construction. Theorem 3.1 shows such an equivalence. **Section 3.2** is dedicated to study the Goppa codes to which we apply our construction, getting Theorem 3.3, showing that the amalgamation of a Goppa code is also a Goppa code. **Section 3.3** is devoted to study the dual of the amalgamated code. We obtain Proposition 3.5 and Theorem 3.4. In **Section 3.4** we generalize Proposition 3.6 and Theorem 3 from [14] when q is prime such that $q \equiv 3 \pmod 4$. Proposition 3.8 and Theorem 3.5 show such generalization in this $q-$ary case. **Section 3.5** contains the main results from this chapter. We discuss various applications. First, we obtain a family of two-weight codes, see Table 3–9 and Theorem 3.6. Second, from a binary two-weight code C and applying our technique to it, we get a code over $\mathbb{F}_4$ which is almost six-weight. See Theorem 3.7. When C is the first-order generalized Reed-Muller code we get a three-weight code over $\mathbb{F}_{q^2}$, see Theorem 3.8 , its Corollary 3.2 and, Theorem 3.10.

3.1 THE GO-UP CONSTRUCTION

We begin this section defining a **Go-Up** construction, we call it amalgamation. We recall that

$$\mathbb{F}_{q^m} = \mathbb{F}_q[x]/(p(x)),$$

where $p(x) \in \mathbb{F}_q[x]$ is a degree m monic irreducible polynomial over F_q. We recall from Galois theory, that a finite field with q^m elements is unique up to isomorphism. However, the multiplicative and additive structures depend upon the specific irreducible or primitive polynomials $p(x)$. Hence, the combinatorial structure and the complexity of computations depend on the choice of $p(x)$.

To illustrate this, when $m = 2$, we write $p(x) = x^2 + \beta x + \alpha$ and $\delta \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$ such that $p(\delta) = 0$, that is,

$$\delta^2 = -\alpha - \delta\beta. \tag{3.1}$$

For $\lambda = \lambda_1 + \delta\lambda_2$ and $\theta = \theta_1 + \delta\theta_2$ in $\mathbb{F}_{q^2}$,

$$\begin{aligned} \lambda\theta &= (\lambda_1 + \delta\lambda_2)(\theta_1 + \delta\theta_2) = (\lambda_1 + \lambda_2 x)(\theta_1 + \theta_2 x) \pmod{p(x)} \\ &= \lambda_1\theta_1 - \lambda_2\alpha\theta_2 + \delta\left(\lambda_2\theta_1 + (\lambda_1 - \lambda_2\beta)\theta_2\right). \end{aligned}$$

For example, if $p(x) = x^2 + x + 1$, that is, $\alpha = 1$ and $\beta = 1$, we get

$$\lambda\theta = \lambda_1\theta_1 - \lambda_2\theta_2 + \delta\left(\lambda_2\theta_1 + (\lambda_1 - \lambda_2)\theta_2\right).$$

For the binary case, $x^2 + x + 1$ is irreducible over $\mathbb{F}_2$ and we take $\delta \in \mathbb{F}_4$ such that $\delta^2 + \delta + 1 = 0$. Then

$$\lambda\theta = \lambda_1\theta_1 + \lambda_2\theta_2 + \delta\left(\lambda_2\theta_1 + (\lambda_1 + \lambda_2)\theta_2\right).$$

For $\mathbf{a} = (a_1 \ \ldots \ a_n)$, $\mathbf{b} = (b_1 \ \ldots \ b_n)$ in $\mathbb{F}_q^n$, we define

$$\mathbf{a}\hat{+}\mathbf{b} = (a_1 + \delta b_1 \ \ldots \ a_n + \delta b_n) := (a_1 b_1 \ \ldots \ a_n b_n) \in \mathbb{F}_{q^2}^n.$$

For example, $\mathbf{a} = (0\ 1\ 1) \in \mathbb{F}_2^3$ and $\mathbf{b} = (1\ 1\ 0) \in \mathbb{F}_2^3$,

$$\mathbf{a}\hat{+}\mathbf{b} = (01\ 11\ 10) = (\delta\ \delta^2\ 1) \in \mathbb{F}_4^3$$

and

$$\mathbf{b}\hat{+}\mathbf{a} = (10\ 11\ 01) = (1\ \delta^2\ \delta).$$

We write for $\mathbf{a},\ \mathbf{b}$ in $\mathbb{F}_q^n$

$$\mathbf{a}\hat{+}\mathbf{b} = \mathbf{a} + \delta\mathbf{b} \in \mathbb{F}_{q^2}^n.$$

Let $\mathbf{x} = \mathbf{a}\hat{+}\mathbf{b}$, $\mathbf{y} = \mathbf{c}\hat{+}\mathbf{d}$ be elements of $\mathbb{F}_{q^2}^n$ and

$$tr : \mathbb{F}_{q^2} \to \mathbb{F}_q, \quad x \to x + x^q$$

be the trace of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$. If we take δ being a primitive element of $\mathbb{F}_{q^2}$, i.e., $p(x)$ is a primitive polynomial, the set $\{1,\ \delta\}$ is a basis, polynomial basis, of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$ because the matrix

$$A = \begin{pmatrix} 1 & \delta \\ 1 & \delta^q \end{pmatrix}$$

is such that $det(A) = \delta^q - \delta \neq 0$. If $\delta^q - \delta = 0$ then $\delta^{q-1} = 1$ which is not possible, because δ is primitive and $q - 1 < q^2 - 1$.

Observe that the linear operator

$$L_\delta : \mathbb{F}_{q^2} \to \mathbb{F}_{q^2}, \quad x \to \delta x$$

has the matrix

$$[L_\delta] = \begin{pmatrix} 0 & -\alpha \\ 1 & -\beta \end{pmatrix}$$

as a representation over the polynomial basis. Then

$$tr(\delta) = \delta + \delta^q = tr[L_\delta] = -\beta$$

and

$$N(\delta) = \delta\delta^q = \delta^{q+1} = det[L_\delta] = \alpha.$$

Therefore we get:

a. Euclidean inner product

$$\mathbf{x}\cdot\mathbf{y} = \sum_i x_i y_i = (\mathbf{a}+\delta\mathbf{b})\cdot(\mathbf{c}+\delta\mathbf{d}) = (\mathbf{a}\cdot\mathbf{c}-\alpha\mathbf{b}\cdot\mathbf{d})+\delta(\mathbf{b}\cdot\mathbf{c}+\mathbf{a}\cdot\mathbf{d}-\beta\mathbf{b}\cdot\mathbf{d}). \quad (3.2)$$

It is bilinear over $\mathbb{F}_{q^2}$ and symmetric. By linearity, for any subset C of $\mathbb{F}_{q^2}^n$,

$$C^\perp = \left\{\boldsymbol{y} = \boldsymbol{c}+\delta\boldsymbol{d} \in \mathbb{F}_{q^2}^n \mid \boldsymbol{x}\cdot\boldsymbol{y} = 0 \ \ \forall \boldsymbol{x} \in C\right\}$$

is linear over $\mathbb{F}_{q^2}$.

b. Trace Euclidean inner product

$$\mathbf{x}\cdot_T\mathbf{y} = tr[(\mathbf{a}+\delta\mathbf{b})\cdot(\mathbf{c}+\delta\mathbf{d})] = 2(\mathbf{a}\cdot\mathbf{c}-\alpha\mathbf{b}\cdot\mathbf{d}) + (\delta+\delta^q)(\mathbf{b}\cdot\mathbf{c}+\mathbf{a}\cdot\mathbf{d}-\beta\mathbf{b}\cdot\mathbf{d}).$$

That is,

$$\mathbf{x}\cdot_T\mathbf{y} = 2(\mathbf{a}\cdot\mathbf{c}-\alpha\mathbf{b}\cdot\mathbf{d}) - \beta(\mathbf{b}\cdot\mathbf{c}+\mathbf{a}\cdot\mathbf{d}-\beta\mathbf{b}\cdot\mathbf{d}). \quad (3.3)$$

In general, it is not linear over $\mathbb{F}_{q^2}$ because the trace is not. But, since

$$\boldsymbol{x}\cdot_T(\delta\boldsymbol{y}) = tr(\boldsymbol{x}\cdot(\delta\boldsymbol{y})) = tr((\delta\boldsymbol{x})\cdot\boldsymbol{y}) = \delta\boldsymbol{x}\cdot_T\boldsymbol{y},$$

if $C \subset \mathbb{F}_{q^2}^n$ is linear over $\mathbb{F}_{q^2}$, then

$$C^{\perp_T} = \left\{\boldsymbol{y} = \boldsymbol{c}+\delta\boldsymbol{d} \in \mathbb{F}_{q^2}^n \mid \boldsymbol{x}\cdot_T\boldsymbol{y} = 0 \ \ \forall \boldsymbol{x} \in C\right\}$$

is linear over $\mathbb{F}_{q^2}$.

c. Hermitian inner product

$$\begin{aligned}\mathbf{x}\cdot_H\mathbf{y} &= \sum_i x_i\overline{y_i} = (\mathbf{a}+\delta\mathbf{b})\cdot(\mathbf{c}+\delta^q\mathbf{d}) = \mathbf{a}\cdot\mathbf{c}+\delta^q\mathbf{a}\cdot\mathbf{d}+\delta\mathbf{b}\cdot\mathbf{c}+\delta^{q+1}\mathbf{b}\cdot\mathbf{d} \\ &= \mathbf{a}\cdot\mathbf{c}+(\delta^q+\delta-\delta)\mathbf{a}\cdot\mathbf{d}+\delta\mathbf{b}\cdot\mathbf{c}+\alpha\mathbf{b}\cdot\mathbf{d}.\end{aligned}$$

That is,

$$\mathbf{x} \cdot_H \mathbf{y} = (\mathbf{a} + \delta\mathbf{b}) \cdot (\mathbf{c} + \delta^q\mathbf{d}) = (\mathbf{a} \cdot \mathbf{c} + \alpha\mathbf{b} \cdot \mathbf{d} - \beta\mathbf{a} \cdot \mathbf{d}) + \delta(\mathbf{b} \cdot \mathbf{c} - \mathbf{a} \cdot \mathbf{d}). \quad (3.4)$$

Since $(\delta\boldsymbol{x}) \cdot_H \boldsymbol{y} = \delta(\boldsymbol{x} \cdot_H \boldsymbol{y})$ and $\boldsymbol{x} \cdot_H (\delta\boldsymbol{y}) = \overline{\delta}(\boldsymbol{x} \cdot_H \boldsymbol{y}) = \delta^q(\boldsymbol{x} \cdot_H \boldsymbol{y})$, "$\cdot_H$" is linear in the first coordinate and has conjugate linearity in the second. From the conjugate linearity we get that for any subset C of $\mathbb{F}_{q^2}^n$

$$C^{\perp_H} = \left\{\boldsymbol{y} = \boldsymbol{c} + \delta\boldsymbol{d} \in \mathbb{F}_{q^2}^n \mid \boldsymbol{x} \cdot_H \boldsymbol{y} = 0 \ \ \forall \boldsymbol{x} \in C\right\}$$

is linear over $\mathbb{F}_{q^2}$.

d. Trace Hermitian inner product

$$\mathbf{x} \cdot_{TH} \mathbf{y} = tr[\mathbf{x} \cdot_H \mathbf{y}] = 2(\mathbf{a} \cdot \mathbf{c} + \delta^{q+1}\mathbf{b} \cdot \mathbf{d}) + (\delta + \delta^q)(\mathbf{b} \cdot \mathbf{c} - \mathbf{a} \cdot \mathbf{d}).$$

That is,

$$\mathbf{x} \cdot_{TH} \mathbf{y} = 2(\mathbf{a} \cdot \mathbf{c} + \alpha\mathbf{b} \cdot \mathbf{d}) + \beta(\mathbf{a} \cdot \mathbf{d} - \mathbf{b} \cdot \mathbf{c}). \quad (3.5)$$

It is not linear over $\mathbb{F}_{q^2}$ by property of the trace. But, since $\boldsymbol{x} \cdot_{TH} (\delta\boldsymbol{y}) = tr(\boldsymbol{x} \cdot_H (\delta\boldsymbol{y})) = tr((\overline{\delta}\boldsymbol{x}) \cdot_H \boldsymbol{y}) = (\overline{\delta}\boldsymbol{x}) \cdot_{TH} \boldsymbol{y}$, if $C \subset \mathbb{F}_{q^2}^n$ is linear over $\mathbb{F}_{q^2}$ then

$$C^{\perp_{TH}} = \left\{\boldsymbol{y} = \boldsymbol{c} + \delta\boldsymbol{d} \in \mathbb{F}_{q^2}^n \mid \boldsymbol{x} \cdot_{TH} \boldsymbol{y} = 0 \ \ \forall \boldsymbol{x} \in C\right\}$$

is linear over $\mathbb{F}_{q^2}$.

Definition 3.1. *Let q be a power of a prime number p, take linear codes C_i in $\mathbb{F}_q^n$ with $0 \leq i \leq m-1$, and $\{1, \delta, \cdots, \delta^{m-1}\}$ a polynomial basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. We define the* **Go-Up** *code over $\mathbb{F}_{q^m}$, denoted $\mathbf{GU}(C_0, C_1, \cdots, C_{m-1})$, by setting it equal to*

$$\left\{\boldsymbol{a}_0 + \delta\boldsymbol{a}_1 + \cdots + \delta^{m-1}\boldsymbol{a}_{m-1} \mid \boldsymbol{a}_i \in C_i, \ \ 0 \leq i \leq m-1\right\}.$$

This definition is in a sense gluing or amalgamating the m codes $C_i's$. When all the m codes are the same code C_0, we denote the Go-Up code by $\mathbf{GU}(m, C_0)$.

Example 3.1. *We take the dual of the binary Hamming code, that is,* C_0 *is the simplex code with parameters* $[7,\ 3,\ 4]_2$. *The code* $\mathbf{GU}(2,\ C_0) = \mathbf{GU}(2,\ S_3(2))$ *is given by Table 3–2.*

$C_0 = S_3(2)$			
(0 0 0 0 0 0 0)	(0 1 1 1 1 0 0)	(1 0 1 1 0 1 0)	(1 1 0 1 0 0 1)
(1 1 0 0 1 1 0)	(1 0 1 0 1 0 1)	(0 1 1 0 0 1 1)	(0 0 0 1 1 1 1)

Table 3–1

$C_0 \hat{+} C_0$			
(0 0 0 0 0 0 0)	(0 δ δ δ δ 0 0)	(δ 0 δ δ 0 δ 0)	(δ δ 0 δ 0 0 δ)
(δ δ 0 0 δ δ 0)	(δ 0 δ 0 δ 0 δ)	(0 δ δ 0 0 δ δ)	(0 0 0 δ δ δ δ)
(0 1 1 1 1 0 0)	(0 δ^2 δ^2 δ^2 δ^2 0 0)	(δ 1 δ^2 δ^2 1 δ 0)	(δ δ^2 1 δ^2 1 0 δ)
(δ δ^2 1 1 δ^2 δ 0)	(δ 1 δ^2 1 δ^2 0 δ)	(0 δ^2 δ^2 1 1 δ δ)	(0 1 1 δ^2 δ^2 δ δ)
(1 0 1 1 0 1 0)	(1 δ δ^2 δ^2 δ 1 0)	(δ^2 0 δ^2 δ^2 0 δ^2 0)	(δ^2 δ 1 δ^2 0 1 δ)
(δ^2 δ 1 1 δ δ^2 0)	(δ^2 0 δ^2 1 δ 1 δ)	(1 δ δ^2 1 0 δ^2 δ)	(1 0 1 δ^2 δ δ^2 δ)
(1 1 0 1 0 0 1)	(1 δ^2 δ δ^2 δ 0 1)	(δ^2 1 δ δ^2 0 δ 1)	(δ^2 δ^2 0 δ^2 0 0 δ^2)
(δ^2 δ^2 0 1 δ δ 1)	(δ^2 1 δ 1 δ 0 δ^2)	(1 δ^2 δ 1 0 δ δ^2)	(1 1 0 δ^2 δ δ δ^2)
(1 1 0 0 1 1 0)	(1 δ^2 δ δ δ^2 1 0)	(δ^2 1 δ δ 1 δ^2 0)	(δ^2 δ^2 0 δ 1 1 δ)
(δ^2 δ^2 0 0 δ^2 δ^2 0)	(δ^2 1 δ 0 δ^2 1 δ)	(1 δ^2 δ 0 1 δ^2 δ)	(1 1 0 δ δ^2 δ^2 δ)
(1 0 1 0 1 0 1)	(1 δ δ^2 δ δ^2 0 1)	(δ^2 0 δ^2 δ 1 δ 1)	(δ^2 δ 1 δ 1 0 δ^2)
(δ^2 δ 1 0 δ^2 δ 1)	(δ^2 0 δ^2 0 δ^2 0 δ^2)	(1 δ δ^2 0 1 δ δ^2)	(1 0 1 δ δ^2 δ δ^2)
(0 1 1 0 0 1 1)	(0 δ^2 δ^2 δ δ 1 1)	(δ 1 δ^2 δ 0 δ^2 1)	(δ δ^2 1 δ 0 1 δ^2)
(δ δ^2 1 0 δ δ^2 1)	(δ 1 δ^2 0 δ 1 δ^2)	(0 δ^2 δ^2 0 0 δ^2 δ^2)	(0 1 1 δ δ δ^2 δ^2)
(0 0 0 1 1 1 1)	(0 δ δ δ^2 δ^2 1 1)	(δ 0 δ δ^2 1 δ^2 1)	(δ δ 0 δ^2 1 1 δ^2)
(δ δ 0 1 δ^2 δ^2 1)	(δ 0 δ 1 δ^2 1 δ^2)	(0 δ δ 1 1 δ^2 δ^2)	(0 0 0 δ^2 δ^2 δ^2 δ^2)

Table 3–2: $\mathbf{GU}(2,\ S_3(2))$

We observe that $\mathbf{GU}(2,\ S_3(2))$ is a two-weight code and is linear over $\mathbb{F}_4$.

Example 3.2. *Let $C_0 = S_3(2)$ and $C_1 = H_3(2)$, where we use the following matrix as a parity check matrix for the Hamming code.*

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Since $H_3(2) = S_3(2) \cup (\boldsymbol{a} + S_3(2))$, we have

$$\mathbf{GU}(C_0,\ C_1) = S_3(2) + \delta H_3(2) = (S_3(2) + \delta S_3(2)) \cup (S_3(2) + \delta(\boldsymbol{a} + S_3(2)))$$

Taking $\boldsymbol{a} = (1\ 1\ 1\ 1\ 0\ 0\ 0\ 0)$, Table 3–3 gives the elements of $\boldsymbol{a} + S_3(2)$.

$\boldsymbol{a} + S_3(2)$			
(1 1 1 0 0 0 0)	(1 0 0 1 1 0 0)	(0 1 0 1 0 1 0)	(0 0 1 1 0 0 1)
(0 0 1 0 1 1 0)	(0 1 0 0 1 0 1)	(1 0 0 0 0 1 1)	(1 1 1 1 1 1 1)

Table 3–3: $\boldsymbol{a} + S_3(2)$

The elements of $S_3(2) + \delta S_3(2)$ are given in Table 3–2 and the elements of $S_3(2) + \delta(\boldsymbol{a} + S_3(2))$ are given in Table 3–4.

$S_3(2) + \delta(\boldsymbol{a} + S_3(2))$			
$(\delta\ \delta\ \delta\ 0\ 0\ 0\ 0)$	$(\delta\ 0\ 0\ \delta\ \delta\ 0\ 0)$	$(0\ \delta\ 0\ \delta\ 0\ \delta\ 0)$	$(0\ 0\ \delta\ \delta\ 0\ 0\ \delta)$
$(0\ 0\ \delta\ 0\ \delta\ \delta\ 0)$	$(0\ \delta\ 0\ 0\ \delta\ 0\ \delta)$	$(\delta\ 0\ 0\ 0\ 0\ \delta\ \delta)$	$(\delta\ \delta\ \delta\ \delta\ \delta\ \delta\ \delta)$
$(\delta\ \delta^2\ \delta^2\ 1\ 1\ 0\ 0)$	$(\delta\ 1\ 1\ \delta^2\ \delta^2\ 0\ 0)$	$(0\ \delta^2\ 1\ \delta^2\ 1\ \delta\ 0)$	$(0\ 1\ \delta^2\ \delta^2\ 1\ 0\ \delta)$
$(0\ 1\ \delta^2\ 1\ \delta^2\ \delta\ 0)$	$(0\ \delta^2\ 1\ 1\ \delta^2\ 0\ \delta)$	$(\delta\ 1\ 1\ 1\ 1\ \delta\ \delta)$	$(\delta\ \delta^2\ \delta^2\ \delta^2\ \delta^2\ \delta\ \delta)$
$(\delta^2\ \delta\ \delta^2\ 1\ 0\ 1\ 0)$	$(\delta^2\ 0\ 1\ \delta^2\ \delta\ 1\ 0)$	$(1\ \delta\ 1\ \delta^2\ 0\ \delta^2\ 0)$	$(1\ 0\ \delta^2\ \delta^2\ 0\ 1\ \delta)$
$(1\ 0\ \delta^2\ 1\ \delta\ \delta^2\ 0)$	$(1\ \delta\ 1\ 1\ \delta\ 1\ \delta)$	$(\delta^2\ 0\ 1\ 1\ 0\ \delta^2\ \delta)$	$(\delta^2\ \delta\ \delta^2\ \delta^2\ \delta\ \delta^2\ \delta)$
$(\delta^2\ \delta^2\ \delta\ 1\ 0\ 0\ 1)$	$(\delta^2\ 1\ 0\ \delta^2\ \delta\ 0\ 1)$	$(1\ \delta^2\ 0\ \delta^2\ 0\ \delta\ 1)$	$(1\ 1\ \delta\ \delta^2\ 0\ 0\ \delta^2)$
$(1\ 1\ \delta\ 1\ \delta\ \delta\ 1)$	$(1\ \delta^2\ 0\ 1\ \delta\ 0\ \delta^2)$	$(\delta^2\ 1\ 0\ 1\ 0\ \delta\ \delta^2)$	$(\delta^2\ \delta^2\ \delta\ \delta^2\ \delta\ \delta\ \delta^2)$
$(\delta^2\ \delta^2\ \delta\ 0\ 1\ 1\ 0)$	$(\delta^2\ 1\ 0\ \delta\ \delta^2\ 1\ 0)$	$(1\ \delta^2\ 0\ \delta\ 1\ \delta^2\ 0)$	$(1\ 1\ \delta\ \delta\ 1\ 1\ \delta)$
$(1\ 1\ \delta\ 0\ \delta^2\ \delta^2\ 0)$	$(1\ \delta^2\ 0\ 0\ \delta^2\ 1\ \delta)$	$(\delta^2\ 1\ 0\ 0\ 1\ \delta^2\ \delta)$	$(\delta^2\ \delta^2\ \delta\ \delta\ \delta^2\ \delta^2\ \delta)$
$(\delta^2\ \delta\ \delta^2\ 0\ 1\ 0\ 1)$	$(\delta^2\ 0\ 1\ \delta\ \delta^2\ 0\ 1)$	$(1\ \delta\ 1\ \delta\ 1\ \delta\ 1)$	$(1\ 0\ \delta^2\ \delta\ 1\ 0\ \delta^2)$
$(1\ 0\ \delta^2\ 0\ \delta^2\ \delta\ 1)$	$(1\ \delta\ 1\ 0\ \delta^2\ 0\ \delta^2)$	$(\delta^2\ 0\ 1\ 0\ 1\ \delta\ \delta^2)$	$(\delta^2\ \delta\ \delta^2\ \delta\ \delta^2\ \delta\ \delta^2)$
$(\delta\ \delta^2\ \delta^2\ 0\ 0\ 1\ 1)$	$(\delta\ 1\ 1\ \delta\ \delta\ 1\ 1)$	$(0\ \delta^2\ 1\ \delta\ 0\ \delta^2\ 1)$	$(0\ 1\ \delta^2\ \delta\ 0\ 1\ \delta^2)$
$(0\ 1\ \delta^2\ 0\ \delta\ \delta^2\ 1)$	$(0\ \delta^2\ 1\ 0\ \delta\ 1\ \delta^2)$	$(\delta\ 1\ 1\ 0\ 0\ \delta^2\ \delta^2)$	$(\delta\ \delta^2\ \delta^2\ \delta\ \delta\ \delta^2\ \delta^2)$
$(\delta\ \delta\ \delta\ 1\ 1\ 1\ 1)$	$(\delta\ 0\ 0\ \delta^2\ \delta^2\ 1\ 1)$	$(0\ \delta\ 0\ \delta^2\ 1\ \delta^2\ 1)$	$(0\ 0\ \delta\ \delta^2\ 1\ 1\ \delta^2)$
$(0\ 0\ \delta\ 1\ \delta^2\ \delta^2\ 1)$	$(0\ \delta\ 0\ 1\ \delta^2\ 1\ \delta^2)$	$(\delta\ 0\ 0\ 1\ 1\ \delta^2\ \delta^2)$	$(\delta\ \delta\ \delta\ \delta^2\ \delta^2\ \delta^2\ \delta^2)$

Table 3–4: $S_3(2) + \delta(\boldsymbol{a} + S_3(2))$

Finally, the $2^7 = 128$ *codewords of* $\mathbf{GU}(S_3(2), H_3(2))$ *are given by Tables 3–2 and 3–4. Observe that for any* $\boldsymbol{x} = \boldsymbol{a} + \delta\boldsymbol{b}$ *in* $S_3(2) + \delta H_3(2)$*, we get* $\omega(\boldsymbol{x}) \in \{3,\ 4,\ 5,\ 6,\ 7\}$.

Example 3.3. *Take* $C_0 = \mathbb{F}_3^2 = \{(0\ 0), (1\ 0), (2\ 0), (1\ 1), (2\ 2), (0\ 1), (0\ 2), (1\ 2), (2\ 1)\}$ *and* $C_1 = \{(0\ 0), (1\ 1), (2\ 2)\}$*, where* $p(x) = x^2 - x - 1$ *is irreducible over* $\mathbb{F}_3$ *and* $\delta^2 = \delta + 1$*, i.e.,* $\{1,\ \delta\}$ *is a polynomial basis of* $\mathbb{F}_{3^2}$ *over* $\mathbb{F}_3$ *. Then,* $\mathbf{GU}(C_0, C_1)$

$\mathbf{GU}(C_0, C_1)$				
$(0\ 0)$	$(1\ 0)$	$(2\ 0)$	$(1\ 1)$	$(2\ 2)$
$(0\ 1)$	$(0\ 2)$	$(1\ 2)$	$(2\ 1)$	$(\delta\ \delta)$
$(\delta^2\ \delta)$	$(2+\delta\ \delta)$	$(\delta^2\ \delta^2)$	$(2+\delta\ 2+\delta)$	$(\delta\ \delta^2)$
$(\delta\ 2+\delta)$	$(\delta^2\ 2+\delta)$	$(2+\delta\ \delta^2)$	$(2\delta\ 2\delta)$	$(1+2\delta\ 2\delta)$
$(2\delta^2\ 2\delta)$	$(1+2\delta\ 1+2\delta)$	$(2\delta^2\ 2\delta^2)$	$(2\delta\ 1+2\delta)$	$(2\delta\ 2\delta^2)$
$(1+2\delta\ 2\delta^2)$	$(2\delta^2\ 1+2\delta)$			

Table 3–5

is an additive code over $\mathbb{F}_{3^2}$*. We observe that a generator matrix of* C_0 *is*

$$G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and generator matrix over $\mathbb{F}_3$ *of* $\mathbf{GU}(C_0, C_1)$ *is*

$$G_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \delta & \delta \end{pmatrix}.$$

Example 3.4. *Consider* $C_0 = S_2(3)$*, the simplex code of dimension* 2 *over* $\mathbb{F}_3$*,* $p(x) = x^2 - x - 1$ *an irreducible polynomial over* $\mathbb{F}_3$ *and* $\delta \in \mathbb{F}_{3^2}$ *such that* $\delta^2 = \delta + 1$*. Its elements are shown in Table 3–6.*

$C_0 = S_2(3)$		
$(0\ 0\ 0\ 0)$	$(1\ 0\ 1\ 1)$	$(2\ 0\ 2\ 2)$
$(0\ 1\ 1\ 2)$	$(0\ 2\ 2\ 1)$	$(1\ 1\ 2\ 0)$
$(1\ 2\ 0\ 2)$	$(2\ 1\ 0\ 1)$	$(2\ 2\ 1\ 0)$

Table 3–6

Let $C_1 = \{(0\ 0\ 0\ 0), (1\ 0\ 1\ 1), (2\ 0\ 2\ 2)\}$ *be a subcode of* $S_2(3)$. *Table 3–7 shows the elements of the additive code* $\mathbf{GU}(C_0, C_1)$ *over* $\mathbb{F}_{3^2}$.

$\mathbf{GU}(C_0, C_1)$			
$(0\ 0\ 0\ 0)$	$(1\ 0\ 1\ 1)$	$(2\ 0\ 2\ 2)$	$(0\ 1\ 1\ 2)$
$(0\ 2\ 2\ 1)$	$(1\ 1\ 2\ 0)$	$(1\ 2\ 0\ 2)$	$(2\ 1\ 0\ 1)$
$(2\ 2\ 1\ 0)$	$(\delta\ 0\ \delta\ \delta)$	$(\delta^2\ 0\ \delta^2\ \delta^2)$	$(2+\delta\ 0\ 2+\delta\ 2+\delta)$
$(\delta\ 1\ \delta^2\ 2+\delta)$	$(\delta\ 2\ 2+\delta\ \delta^2)$	$(\delta^2\ 1\ 2+\delta\ \delta)$	$(\delta^2\ 2\ \delta\ 2+\delta)$
$(2+\delta\ 1\ \delta\ \delta^2)$	$(2+\delta\ 2\ \delta^2\ \delta)$	$(2\delta\ 0\ 2\delta\ 2\delta)$	$(1+2\delta\ 0\ 1+2\delta\ 1+2\delta)$
$(2\delta^2\ 0\ 2\delta^2\ 2\delta^2)$	$(2\delta\ 1\ 1+2\delta\ 2\delta^2)$	$(2\delta\ 2\ 2\delta^2\ 1+2\delta)$	$(1+2\delta\ 1\ 2\delta^2\ 2\delta)$
$(1+2\delta\ 2\ 2\delta\ 2\delta^2)$	$(2\delta^2\ 1\ 2\delta\ 1+2\delta)$	$(2\delta^2\ 2\ 1+2\delta\ 2\delta)$	

Table 3–7

We observe that a generator matrix of $S_2(3)$ *is given by*

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix}$$

and then a generator matrix of the additive code $\mathbf{GU}(C_0, C_1)$ *(this code is linear over* $\mathbb{F}_3$*) is given by*

$$G_1 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ \delta & 0 & \delta & \delta \end{pmatrix}.$$

We observe that $\mathbf{GU}(C_0, C_1)$ *is an additive two-weight code.*

Lemma 3.1. $\mathbf{GU}(C_0, \cdots, C_{m-1})$ *is additive over* $\mathbb{F}_{q^m}$ *with minimum distance*

$$d = min\,\{d_0,\ \cdots,\ d_{m-1}\}$$

and $q^{(k_0+\cdots+k_{m-1})}$ *codewords over* $\mathbb{F}_q$. *In addition,* $\mathbf{GU}(C_0, \cdots, C_{m-1})$ *is linear over* $\mathbb{F}_{q^m}$ *if and only if* $\mathbf{GU}(C_0, \cdots, C_{m-1}) = \mathbf{GU}(m,\ C_0)$ *and if* C_0 *is an* $[n,\ k_0,\ d_0]_q$ *linear code, then* $\mathbf{GU}(m,\ C_0)$ *is an* $[n,\ k_0,\ d_0]_{q^m}$ *linear code.*

Proof. Given $\boldsymbol{x} \in \mathbf{GU}(C_0, \cdots, C_{m-1})$,

$$\boldsymbol{x} = \sum_{i=0}^{m-1} \delta^i \boldsymbol{a}_i = \left(\sum_{i=0}^{m-1} \delta^i \boldsymbol{a}_{i1} \ \cdots \ \sum_{i=0}^{m-1} \delta^i \boldsymbol{a}_{in} \right) \tag{3.6}$$

That is

$$\boldsymbol{x} = \begin{pmatrix} 1 & \delta & \cdots & \delta^{m-1} \end{pmatrix} \begin{pmatrix} a_{01} & \cdots & a_{0n} \\ a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{(m-1)1} & \cdots & a_{(m-1)n} \end{pmatrix}. \tag{3.7}$$

Then, we can see that the weight of $\boldsymbol{x}$ is the number of nonzero columns of the $m \times n$ matrix in Equation (3.7), that is, the minimum distance of $\mathbf{GU}(C_0, \cdots, C_{m-1})$ is

$$d = min\left\{d_0, \ \cdots, \ d_{m-1}\right\},$$

where d_i is the minimum distance of C_i.

Now, for $\boldsymbol{x} = \sum_{i=0}^{m-1} \delta^i \boldsymbol{a}_i$ and $\boldsymbol{y} = \sum_{i=0}^{m-1} \delta^i \boldsymbol{b}_i$ in $\mathbf{GU}(C_0, \cdots, C_{m-1})$, from Equation (3.6), we get that $\boldsymbol{x}+\boldsymbol{y} = \sum_{i=0}^{m-1} \delta^i (\boldsymbol{a}_i + \boldsymbol{b}_i) \in \mathbf{GU}(C_0, \cdots, C_{m-1})$. Then $\mathbf{GU}(C_0, \cdots, C_{m-1})$ is an additive code over $\mathbb{F}_{q^m}$.

On the other hand, let $p(x) = x^m + r_{m-1}x^{m-1} + \cdots + r_1 x + r_0$ be an irreducible polynomial of degree m over $\mathbb{F}_q$ such that $p(\delta) = 0$, that is, $\delta^m = -r_0 - r_1\delta - \cdots - r_{m-1}\delta^{m-1}$. Then

$$\begin{aligned} \delta\boldsymbol{x} &= \sum_{i=0}^{m-1} \delta^{i+1} \boldsymbol{a}_i = \sum_{i=0}^{m-2} \delta^{i+1} \boldsymbol{a}_i - \left(\sum_{i=0}^{m-1} \delta^i r_i \right) \boldsymbol{a}_{m-1} & (3.8) \\ &= (-r_0 \boldsymbol{a}_{m-1}) + \delta\left(\boldsymbol{a}_0 - r_1 \boldsymbol{a}_{m-1}\right) + \cdots + \delta^{m-1}\left(\boldsymbol{a}_{m-2} - r_{m-1}\boldsymbol{a}_{m-1}\right). & (3.9) \end{aligned}$$

If $\delta\boldsymbol{x} \in \mathbf{GU}(C_0, \cdots, C_{m-1})$, we get that $\boldsymbol{a}_{m-1} \in C_0$, i.e., $C_{m-1} \subset C_0$. Since $\boldsymbol{a}_0 - r_1\boldsymbol{a}_{m-1} \in C_1$, then $C_0 \subset C_1$. Continuing, we obtain that $\boldsymbol{a}_{m-2} - r_{m-1}\boldsymbol{a}_{m-1} \in$

C_{m-1}, i.e., $C_{m-2} \subset C_{m-1}$. That is, $C_{m-1} \subset C_0 \subset C_1 \subset C_2 \subset \cdots \subset C_{m-2} \subset C_{m-1}$. Thus, $C_0 = C_1 = C_2 = \cdots = C_{m-2} = C_{m-1}$ and $\mathbf{GU}(C_0, \cdots, C_{m-1}) = \mathbf{GU}(m,\ C_0)$.

If $\boldsymbol{x} \in \mathbf{GU}(m,\ C_0)$, from Equation (3.8), $\delta\boldsymbol{x} \in \mathbf{GU}(m,\ C_0)$. Then $\mathbf{GU}(m,\ C_0)$ is linear over $\mathbb{F}_{q^m}$ and if C_0 is an $[n,\ k_0,\ d_0]_q$ linear code, then $|\mathbf{GU}(m,\ C_0)| = q^{k_0}q^{k_0}\cdots q^{k_0} = (q^m)^{k_0}$, i.e., $dim_{\mathbb{F}_{q^m}}(\mathbf{GU}(m,\ C_0)) = k_0$. □

In most quantum code constructions, only additivity of the constituent codes is needed. Indeed, for the construction of stabilizer codes, we just need additive codes. The theory of additive codes is much richer than that of linear codes. Linearity gives us linear quantum codes for $m = 2$. Therefore Lemma 3.1 is important for the construction of stabilizer codes.

Example 3.5. *Let C_0 be the repetition code with parameters $[2^m,\ 1,\ 2^m]$ and C_1 be the first-order Reed-Muller code with parameters $[2^m,\ m+1,\ 2^{m-1}]$. Then $\mathbf{GU}(C_0, C_1)$ is an additive code over $\mathbb{F}_4$ with parameters $(2^m,\ 2^{m+2},\ 2^{m-1})$.*

Example 3.6a. *Let $C_0 = S_m(2)$ be the binary simplex code with parameters $[2^m - 1,\ m,\ 2^{m-1}]$ and C_1 the code obtained by puncturing $\mathbf{R}(1,\ m)$, i.e., C_1 has parameters $[2^m - 1,\ m+1,\ 2^{m-1} - 1]$. The additive code over $\mathbb{F}_4$ given by $\mathbf{GU}(S_m(2), C_1)$ has parameters $(2^m - 1,\ 2^{2m+1},\ 2^{m-1} - 1)$.*

b. *Let C_0 be obtained extending the Hamming code $H_m(2)$, i.e., C_0 has parameters $[2^m,\ 2^m - m - 1,\ 4]$ and let $C_1 = \mathbf{R}(1,\ m)$. The additive code over $\mathbb{F}_4$ given by $\mathbf{GU}(C_0, C_1)$ has parameters $(2^m,\ 2^{2^m},\ 4)$ for all $m \geq 3$.*

Observe *that when $m = 3$, we obtain an additive quaternary code with optimal parameters, i.e., we get that $\mathbf{GU}(C_0, C_1)$ has parameters $(8,\ 2^8,\ 4)$. See Table 1 in [8].*

Proposition 3.1. *If $\beta_0 = \{\mathbf{a_1}, \cdots, \mathbf{a_{k_0}}\}$ is a basis of C_0 over $\mathbb{F}_q$, the sets $\beta_1 = \{\mathbf{a_1}, \cdots, \mathbf{a_{k_0}}\}$ and $\beta_2 = \{\mathbf{0}\hat{+}\mathbf{a_1}, \cdots, \mathbf{0}\hat{+}\mathbf{a_{k_0}}\}$ are bases of $\mathbf{GU}(2,\ C_0)$ over $\mathbb{F}_{q^2}$.*

Proof. Consider $\alpha_i = (r_i, s_i) = r_i + \delta s_i \in \mathbb{F}_{q^2}$, $1 \leq i \leq k_0$, such that

$$\sum_i \alpha_i(\mathbf{a_i}\hat{+}\mathbf{0}) = \mathbf{0}.$$

Since $\alpha_i(\mathbf{a_i}\hat{+}\mathbf{0}) = (r_i + \delta s_i)(\mathbf{a_i} + \delta\mathbf{0}) = r_i\mathbf{a_i} + \delta s_i\mathbf{a_i}$, we get

$$\mathbf{0} = \sum_i \alpha_i \left(\mathbf{a_i}\hat{+}\mathbf{0}\right) = \sum_i (r_i\mathbf{a_i} + \delta s_i\mathbf{a_i}) = \sum_i r_i\mathbf{a_i} + \delta \sum_i s_i\mathbf{a_i}.$$

But, $\sum_i r_i\mathbf{a_i} = (x_1 \ \cdots \ x_n) \in \mathbb{F}_q^n$ and $\sum_i s_i\mathbf{a_i} = (y_1 \ \cdots \ y_n) \in \mathbb{F}_q^n$, imply that

$$\mathbf{0} = \sum_i \alpha_i(\mathbf{a_i}\hat{+}\mathbf{0}) = (x_1 + \delta y_1 \ \cdots \ x_n + \delta y_n).$$

Since $x_j \in \mathbb{F}_q$, $y_j \in \mathbb{F}_q$ for $1 \leq j \leq n$, and $\delta \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Then, $x_j = 0$ and $y_j = 0$ for all j, that is, $\sum_i r_i\mathbf{a_i} = 0$ and $\sum_i s_i\mathbf{a_i} = 0$. From this we get $r_i = 0$, $\forall i$ and $s_i = 0$ $\forall i$. Thus, $\alpha_i = 0$, $\forall i$ and β_1 **is linear independent over** $\mathbb{F}_{q^2}$. Now, given any $\mathbf{a}\hat{+}\mathbf{b} \in \mathbf{GU}(2,\ C_0)$, there are scalars r_i and s_i, $1 \leq i \leq k_0$, in $\mathbb{F}_q$ such that

$$\mathbf{a} = \sum_i r_i\mathbf{a_i} \ \textit{and} \ \mathbf{b} = \sum_i s_i\mathbf{a_i}$$

because β_0 is a basis of C_0 over $\mathbb{F}_q$.

Then,

$$\mathbf{a}\hat{+}\mathbf{b} = \sum_i r_i\mathbf{a_i}\hat{+}\sum_i s_i\mathbf{a_i} = \sum_i (r_i\mathbf{a_i}\hat{+}s_i\mathbf{a_i}) = \sum_i (r_i, s_i)(\mathbf{a}\hat{+}\mathbf{0}).$$

From this, there are scalars $\alpha_i = r_i + \delta s_i$, $1 \leq i \leq k_0$, in $\mathbb{F}_{q^2}$ such that

$$\mathbf{a}\hat{+}\mathbf{b} = \sum_i \alpha_i(\mathbf{a_i}\hat{+}\mathbf{0}).$$

Thus, β_1 is a basis of $\mathbf{GU}(2,\ C_0)$ over $\mathbb{F}_{q^2}$.

On the other hand, if there are scalars $\alpha_i = (r_i, s_i) = r_i + \delta s_i \in \mathbb{F}_{q^2}$, $1 \leq i \leq k_0$, such that

$$\sum_i \alpha_i(\mathbf{0}\hat{+}\mathbf{a_i}) = \mathbf{0}$$

Since $\alpha_i(\mathbf{0}\hat{+}\mathbf{a_i}) = (r_i + \delta s_i)\,(\mathbf{0} + \delta\mathbf{a_i}) = -\alpha s_i\mathbf{a_i} + \delta(r_i - s_i\beta)\mathbf{a_i}$,

$$\mathbf{0} = \sum_i \alpha_i(\mathbf{0}\hat{+}\mathbf{a_i}) = \sum_i -\alpha s_i\mathbf{a_i}\hat{+}\sum_i (r_i - s_i\beta)\mathbf{a_i} = \sum_i -\alpha s_i\mathbf{a_i} + \delta\sum_i (r_i - s_i\beta)\mathbf{a_i}.$$

That is, $\sum_i s_i\mathbf{a_i} = \mathbf{0}$ and $\sum_i (r_i - s_i\beta)\mathbf{a_i} = \mathbf{0}$. Then, $s_i = 0$ and $r_i = 0$, $\forall i$. From this, β_2 is linearly independent over $\mathbb{F}_{q^2}$ and we conclude that β_2 is another basis of $\mathbf{GU}(2,\ C_0)$ over $\mathbb{F}_{q^2}$.

□

Remark: From *Proposition* 3.1, if G is a generator matrix of C_0, then G and δG are generator matrices of $\mathbf{GU}(2,\ C_0)$. Observe that the linear code $C_0 + \delta\mathbf{0}$ is the subfield subcode of $\mathbf{GU}(2,\ C_0) = C_0 + \delta C_0$ over $\mathbb{F}_q$.

Proposition 3.2. *If C_0 and C_1 are cyclic codes over $\mathbb{F}_q$, then $C = \mathbf{GU}(C_0,\ C_1)$ is an additive cyclic code over $\mathbb{F}_{q^2}$.*

Proof. Let $\mathbf{a} = (a_1\ \cdots\ a_n) \in C_0$ and $\boldsymbol{b} = (b_1\ \cdots\ b_n) \in C_1$. By definition, $\mathbf{a} + \delta\mathbf{b} = (a_1 + \delta b_1 \cdots a_n + \delta b_n) \in C$ and $(\boldsymbol{a} + \delta\boldsymbol{b})(x) = (a_1 + \delta b_1) + (a_2 + \delta b_2)x + \cdots + (a_n + \delta b_n)x^{n-1} \in \mathbb{F}_{q^2}$. Then

$$\begin{aligned} x(\boldsymbol{a} + \delta\boldsymbol{b})(x) &= (a_1 + \delta b_1)x + \cdots + (a_{n-1} + \delta b_{n-1})x^{n-1} + (a_n + \delta b_n)x^n \\ &= (a_1x + \cdots + a_nx^n) + \delta\left(b_1x + \cdots + b_nx^n\right). \end{aligned}$$

From this,

$$x(\mathbf{a} + \delta\mathbf{b})(x) = xa(x) + \delta xb(x) \in C_0 + \delta C_1$$

because C_0 and C_1 are cyclic codes. □

Let $C_0 \subset \mathbb{F}_q^n$ be a linear code and $C = \mathbf{GU}(m,\ C_0)$. Given $\boldsymbol{x} \in C$, $\boldsymbol{x} = \boldsymbol{a}_0 + \delta\boldsymbol{a}_1 + \cdots + \delta^{m-1}\boldsymbol{a}_{m-1}$, where $\{1,\ \delta,\ \cdots,\ \delta^{m-1}\}$ is an $\mathbb{F}_q$-basis of $\mathbb{F}_{q^m}$ and

$\boldsymbol{a}_i \in C_0$, $0 \leq i \leq m-1$. Then,

$$\boldsymbol{x}^q = \boldsymbol{a}_0 + \delta^q \boldsymbol{a}_1 + \cdots + \delta^{q(m-1)} \boldsymbol{a}_{m-1}.$$

Since $\delta^{qj} \in \mathbb{F}_{q^m}$, with $1 \leq j \leq m-1$, $\delta^{qj} = \alpha_{0j} + \alpha_{1j}\delta + \cdots + \alpha_{(m-1j)}\delta^{m-1}$ for some $\alpha_{ij} \in \mathbb{F}_q$, we get that $\boldsymbol{x}^q \in C$. Thus, $C^q \subset C$, that is, C is Frobenius invariant.

On the other hand, let $C \subset \mathbb{F}_{q^m}^n$ be a linear code and C_0 **its sub-field sub-code** over $\mathbb{F}_q$. From Lemma 3.1, $\mathbf{GU}(m,\ C_0)$ is a linear code over $\mathbb{F}_{q^m}$. Then $\boldsymbol{a}_0 + \delta \boldsymbol{a}_1 + \cdots + \delta^{m-1} \boldsymbol{a}_{m-1}$ is an element of C. Therefore $\mathbf{GU}(m,\ C_0) \subset C$. If in addition $C^q \subset C$, we get that for $\boldsymbol{x} \in C$, $\boldsymbol{x} + \boldsymbol{x}^q + \cdots + \boldsymbol{x}^{q^{m-1}} = tr(\boldsymbol{x}) \in C$. Thus $dim_{\mathbb{F}_q}(tr(C)) \leq dim_{\mathbb{F}_{q^m}}(C)$ and from Delsarte's theorem

$$dim(tr(C)) = dim((C^\perp)_0)^\perp = n - dim(C^\perp)_0 \geq n - dim(C^\perp) = dim(C) \geq dim(C_0).$$

Thus

$$dim_{\mathbb{F}_q}(tr(C)) = dim_{\mathbb{F}_{q^m}}(C) \ \ and \ \ dim_{\mathbb{F}_q}(C_0) = dim_{\mathbb{F}_{q^m}}(C).$$

Also $dim_{\mathbb{F}_{q^m}}(\mathbf{GU}(m,\ C_0)) = dim_{\mathbb{F}_q}(C_0) = k_0$ and we obtain that

$$C = \mathbf{GU}(m,\ C_0).$$

We get then

Lemma 3.2. *Let $C \subset \mathbb{F}_{q^m}^n$ be a Frobenius invariant linear code. Then $C = \mathbf{GU}(m,\ C_0)$, where C_0 is the sub-field sub-code of C over $\mathbb{F}_q$.*

We can summarize the above argument in the following theorem, which gives us a nice relation between **amalgamated codes and Frobenius invariant codes.**

Theorem 3.1. *Given $C \subset \mathbb{F}_{q^m}^n$ and $C^q \subset C$, C is linear over $\mathbb{F}_{q^m}$ if and only if $C = \mathbf{GU}(m,\ C_0)$, where C_0 is the sub-field sub-code of C over $\mathbb{F}_q$. Since C is Frobenius invariant, $tr(C) = \left\{ tr(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{x}^q + \cdots + \boldsymbol{x}^{q^{m-1}} \mid \boldsymbol{x} \in C \right\} = C_0$.*

Now, we will define the amalgamation operation taking a normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. For example, if $\delta \in \mathbb{F}_{q^2}$ is a primitive element, then $\{\delta,\ \delta^q\}$ is a basis

because the matrix

$$A = \begin{pmatrix} \delta & \delta^q \\ \delta^q & \delta^{q^2} \end{pmatrix}$$

is such that $det(A) = \delta\delta - \delta^{2q} \neq 0$. If $\delta^2 - \delta^{2q} = 0$ then $\delta^{2q-2} = 1$, which is not possible because δ is primitive and $2q - 2 < q^2 - 1$.

If we call $p(x) = x^2 + \beta x + \alpha = (x - \delta)(x - \delta^q)$ the minimum polynomial of δ over $\mathbb{F}_q$, we get that $p(x) = x^2 + \beta x + \alpha$ is irreducible over $\mathbb{F}_q$ and $tr(\delta) = \delta + \delta^q = -\beta$ and $N(\delta) = \delta\delta^q = \alpha$.

Definition 3.2. *Let q be a power of a prime number p, take m linear codes C_i in $\mathbb{F}_q^n$, and let $\left\{\delta,\ \delta^q,\ \cdots,\ \delta^{q^{m-1}}\right\}$ be a normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. We define a code over $\mathbb{F}_{q^m}$, denoted $\mathbf{NGU}\,(C_0, \cdots, C_{m-1})$, by*

$$\mathbf{NGU}\,(C_0, \cdots, C_{m-1}) = \left\{\delta\boldsymbol{a}_0 + \delta^q\boldsymbol{a}_1 + \cdots + \delta^{q^{m-1}}\boldsymbol{a}_{m-1} \mid \boldsymbol{a}_i \in C_i,\ \ 0 \leq i \leq m-1\right\}.$$

When all the m codes are the same code C_0 we denote $\mathbf{NGU}\,(C_0, \cdots, C_{m-1})$ by $\mathbf{NGU}\,(m,\ C_0)$.

a. Using a normal basis $\left\{\delta,\ \delta^q,\ \cdots,\ \delta^{q^{m-1}}\right\}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, then

$$\mathbf{NGU}(C_0, \cdots, C_{m-1}) = \left\{\delta\boldsymbol{a}_0 + \delta^q\boldsymbol{a}_1 + \cdots + \delta^{q^{m-1}}\boldsymbol{a}_{m-1} \mid \boldsymbol{a}_i \in C_i\right\},$$

and given $\boldsymbol{x} \in \mathbf{NGU}(C_0, \cdots, C_{m-1})$ we can write

$$\boldsymbol{x} = \sum_{i=0}^{m-1} \delta^{q^{i-1}}\boldsymbol{a}_i = \left(\sum_{i=0}^{m-1} \delta^{q^{i-1}}\boldsymbol{a}_{i1}\ \cdots\ \sum_{i=0}^{m-1} \alpha^{q^{i-1}}\boldsymbol{a}_{in}\right)$$

i.e.,

$$\boldsymbol{x} = \begin{pmatrix} \delta & \delta^q & \cdots & \delta^{q^{m-1}} \end{pmatrix} \begin{pmatrix} a_{01} & \cdots & a_{0n} \\ a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{(m-1)1} & \cdots & a_{(m-1)n} \end{pmatrix}. \tag{3.10}$$

Again, $\mathbf{NGU}(C_0, \cdots, C_{m-1})$ is an additive code over $\mathbb{F}_{q^m}$ and the weight of the codeword $\boldsymbol{x}$ is the number of nonzero columns of the $m \times n$ matrix in Equation (3.10).

We observe that for a normal basis $\{\delta,\ \delta^2\}$ of $\mathbb{F}_4$ over $\mathbb{F}_2$ if C_0 and C_1 are two linear codes in $\mathbb{F}_2^n$, then $\delta\boldsymbol{a} + \delta^2\boldsymbol{b} = \delta(\boldsymbol{a} + \delta\boldsymbol{b})$, that is, in the binary case $\mathbf{NGU}(C_0, C_1) = \delta\mathbf{GU}(C_0, C_1)$.

3.2 THE GO-UP OF A GOPPA CODE

Now, we apply the **Go-Up** construction in the Goppa codes context. We know $\mathbb{F}_4$ is an extension field of $\mathbb{F}_2$ and $\mathbb{F}_{4^m}$ is an extension field of $\mathbb{F}_{2^m}$. We consider $L = \{\alpha_1, \cdots, \alpha_n\} \subseteq \mathbb{F}_{2^m}$ and $g(x) \in \mathbb{F}_{2^m}[x]$ separable polynomial. With the same L and the same $g(x)$, we take $C = \mathbf{\Gamma}(L, g)_{\mathbb{F}_4} \subseteq \mathbb{F}_4^n$. We call $C_0 = \mathbf{\Gamma}(L, g) \subseteq \mathbb{F}_2^n$, and we know, see Theorem 2.10,

$$d_0 \geq 2t + 1,$$

where $t = deg(g(x))$, and $dim_{\mathbb{F}_2} C_0 = k_0 \geq n - mt$. Then $C_1 = \hat{\mathbf{\Gamma}}(L, g) = \mathbf{GU}(2,\ C_0)$ is a linear code over $\mathbb{F}_4$ with the same parameters as the binary Goppa code. That is, it is possible to construct a code over $\mathbb{F}_4$ with the same capability to correct errors as $C_0 = \mathbf{\Gamma}(L, g) \subseteq \mathbb{F}_2^n$.

For $\boldsymbol{a} + \delta\boldsymbol{b} = (a_1 + \delta b_1\ \cdots\ a_n + \delta b_n) \in \hat{\mathbf{\Gamma}}(L, g)$ we have,

$$\sum_i \frac{a_i}{x - \alpha_i} \equiv 0 \pmod{g(x)} \qquad \sum_i \frac{b_i}{x - \alpha_i} \equiv 0 \pmod{g(x)}.$$

Taking m an even number, we have $\boldsymbol{a} + \delta\boldsymbol{b} \in \mathbf{\Gamma}(L, g)_{\mathbb{F}_4}$ because, in this case, $\mathbb{F}_4$ is a subfield of $\mathbb{F}_{2^m}$ and then we may write

$$\sum_i \frac{a_i + \delta b_i}{x - \alpha_i} = \sum_i \frac{a_i}{x - \alpha_i} + \delta \sum_i \frac{b_i}{x - \alpha_i} \equiv\ 0 \pmod{g(x)}.$$

That is,

$$\hat{\mathbf{\Gamma}}(L, g) = \mathbf{GU}\,(2\,; \mathbf{\Gamma}(L, g)_{\mathbb{F}_2}) \subset \mathbf{\Gamma}(L, g)_{\mathbb{F}_4}.$$

We summarize this in the following theorem:

Theorem 3.2. *Given* $L = \{\alpha_1, \cdots, \alpha_n\} \subseteq \mathbb{F}_{2^m}$, $g(x) \in \mathbb{F}_{2^m}[x]$ *and* m *be an even number. Then* $\mathbf{GU}(2\,;\mathbf{\Gamma}(L,g)_{\mathbb{F}_2})$ *is a quaternary linear subcode of* $\mathbf{\Gamma}(L,g(x))_{\mathbb{F}_4}$ *with the same parameters as its subfield subcode.*

With this **Go-Up** construction we obtain a subcode of $\mathbf{\Gamma}(L,g)_{\mathbb{F}_4}$ with the same parameters as $\mathbf{\Gamma}(L,g)_{\mathbb{F}_2}$. The natural question is if it is possible to obtain equality, that is, under what conditions $\hat{\mathbf{\Gamma}}(L,g) = \mathbf{\Gamma}(L,g)$, i.e., when $\mathbf{GU}(2\,;\mathbf{\Gamma}(L,g)_{\mathbb{F}_2})$ is a Goppa code.?

We begin with a particular case, see *Definition* 2.8, taking $\beta \in \mathbb{F}_q$ a primitive *nth* root of unity, $n|(q-1)$ and $\alpha = \beta^{\frac{q-1}{n}}$, let $L_0 = \{\alpha^0, \alpha^1, \cdots, \alpha^{n-1}\} \subset \mathbb{F}_q$. For $k \leq n$, we let

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

and we know

$$\boldsymbol{R}S(n,\ k) = \left\{(f(\alpha^0), f(\alpha^1), \cdots, f(\alpha^{n-1})) \mid f(x) \in P_k\right\} \subset \mathbb{F}_q^n$$

is a cyclic code because for $\boldsymbol{c} = (f(\alpha^0)\ \ f(\alpha^1)\ \cdots\ f(\alpha^{n-1})) \in \boldsymbol{R}S(n,k)$,

$$x\boldsymbol{c} = \left(f(\alpha^{n-1})\ \ f(\alpha^0)\ \cdots\ f(\alpha^{n-2})\right) = \left(f_1(\alpha^0)\ f_1(\alpha^1)\ \cdots\ f_1(\alpha^{n-1})\right),$$

where $f_1(x) = f(\alpha^{-1}x) \in P_k$, i.e., $x\boldsymbol{c} \in \boldsymbol{R}S(n,k)$. We know $\boldsymbol{R}S(n,k)$ is MDS code, i.e., $d-1 = n-k$ and we can also see the $\boldsymbol{R}S(n,k)$ as a Goppa code with polynomial x^{d-1} and support set $L = \left\{\alpha^0, \alpha^{-1}, \cdots, \alpha^{-(n-1)}\right\} \subset \mathbb{F}_q$, i.e., $\boldsymbol{R}S(n,k) = \mathbf{\Gamma}(L, x^{d-1})$, see Proposition 2.2. Then a parity check matrix of $\boldsymbol{R}S(n,k)$ is given by

$$H = \begin{pmatrix} \alpha^{0(d-2)} & \alpha^{-(d-2)} & \cdots & \alpha^{-(n-1)(d-2)} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha^0 & \alpha^{-1} & \cdots & \alpha^{-(n-1)} \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \alpha^0 & 0 & \cdots & 0 \\ 0 & \alpha^{d-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \alpha^{(n-1)(d-1)} \end{pmatrix}.$$

That is,

$$H = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \cdots & \alpha^{(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{d-2} & \alpha^{2(d-2)} & \cdots & \alpha^{(n-1)(d-2)} \\ 1 & \alpha^{d-1} & \alpha^{2(d-1)} & \cdots & \alpha^{(n-1)(d-1)} \end{pmatrix}.$$

On the other hand, if we take a cyclic code C over $\mathbb{F}_q$ of length n with generator polynomial $g(x) = (x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{d-1})$ with $d \geq 2$, a polynomial word $p(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ is such that $p(x) \in C$ if and only if $g(x)|p(x)$ if and only if $p(\alpha^j) = 0$ for $1 \leq j \leq d-1$. That is, $p(x) \in C$ if and only if

$$\begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \cdots & \alpha^{(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{d-2} & \alpha^{2(d-2)} & \cdots & \alpha^{(n-1)(d-2)} \\ 1 & \alpha^{d-1} & \alpha^{2(d-1)} & \cdots & \alpha^{(n-1)(d-1)} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = H(c_0\ c_1\ \cdots\ c_{n-1})^t = 0.$$

Then, $C = \left\{ \boldsymbol{c} \in \mathbb{F}_q^n \mid H\boldsymbol{c}^t = 0 \right\} = \boldsymbol{R}S(n,k)$ with $d - 1 = n - k$. If we take $\boldsymbol{R}S(n,k) \subset \mathbb{F}_{q^2}^n$ and $L_0 = \{\alpha^0, \alpha, \cdots, \alpha^{q-2}\}$, where α is a primitive element for $\mathbb{F}_q^*$, i.e., we take $n = q-1$ and observe that $(q-1)|(q^2-1)$. Then,

$$\boldsymbol{R}S(n,k) = \boldsymbol{\Gamma}(L, x^{d-1}) = \left\{ \boldsymbol{c} \in \mathbb{F}_{q^2}^n \mid \sum_{i=0}^{n-1} \frac{c_i}{x - \alpha^{-i}} \equiv 0 \pmod{x^{d-1}} \right\}.$$

For $\boldsymbol{c} \in \boldsymbol{\Gamma}(L, x^{d-1})$, $\mathbf{c} = \mathbf{a}\hat{+}\mathbf{b} = (a_0 + \delta b_0\ \cdots\ a_{n-1} + \delta b_{n-1})$ for some $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$. From this,

$$\begin{aligned} \sum_{i=0}^{n-1} \frac{c_i}{x - \alpha^{-i}} &= \sum_{i=0}^{n-1} \frac{a_i + \delta b_i}{x - \alpha^{-i}} = \sum_{i=0}^{n-1} \frac{a_i}{x - \alpha^{-i}} + \delta \sum_{i=0}^{n-1} \frac{b_i}{x - \alpha^{-i}} \\ &= \sum_{i=0}^{n-1} a_i p_i(x) + \delta \sum_{i=0}^{n-1} b_i p_i(x), \end{aligned}$$

where $p_i(x) = -\frac{g(x)-g(\alpha^{-i})}{x-\alpha^{-i}} g^{-1}(\alpha^{-i}) \pmod{g(x)}$, i.e., we can see $\frac{1}{x-\alpha^{-i}}$ as a polynomial, over $\mathbf{F}_q$ modulo $g(x) = x^{d-1}$, of degree $d-2$. We write $p_i(x) = a_{0i} + a_{1i}x + \cdots + a_{d-2i}x^{d-2} \in \mathbb{F}_q[x]$ and then

$$\begin{aligned}\sum_{i=0}^{n-1} \frac{c_i}{x-\alpha^{-i}} &= \sum_{i=0}^{n-1} a_i \left(a_{0i} + \cdots + a_{d-2i}x^{d-2}\right) + \delta \sum_{i=0}^{n-1} b_i \left(a_{0i} + \cdots + a_{d-2i}x^{d-2}\right) \\ &= \left(\sum_{i=0}^{n-1} a_i a_{0i} + \delta \sum_{i=0}^{n-1} b_i a_{0i}\right) + \cdots + \left(\sum_{i=0}^{n-1} a_i a_{d-2i} + \delta \sum_{i=0}^{n-1} b_i a_{d-2i}\right) x^{d-2}\end{aligned}$$

Since $\mathbf{c} = \mathbf{a}\hat{+}\mathbf{b} \in \mathbf{\Gamma}(L, x^{d-1})$ and we obtain a polynomial of degree at most $d-2$, then

$$\left(\sum_{i=0}^{n-1} a_i a_{0i} + \delta \sum_{i=0}^{n-1} b_i a_{0i}\right) + \cdots + \left(\sum_{i=0}^{n-1} a_i a_{d-2i} + \delta \sum_{i=0}^{n-1} b_i a_{d-2i}\right) x^{d-2} = 0.$$

That is

$$\sum_{i=0}^{n-1} a_i a_{ji} + \delta \sum_{i=0}^{n-1} b_i a_{ji} = u_{ji} + \delta v_{ji} = 0$$

for $0 \le j \le d-2$, $u_{ji} = \sum_{i=0}^{n-1} a_i a_{ji}$ and $v_{ji} = \sum_{i=0}^{n-1} b_i a_{ji}$. If $v_{ji} \neq 0$, then $\delta = \frac{u_{ji}}{v_{ji}} \in \mathbb{F}_q$ which is not possible because $\delta \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Therefore, $v_{ji} = 0$ and $u_{ji} = 0$ and we conclude that

$$\sum_{i=0}^{n-1} a_i p_i(x) = 0 \text{ and } \sum_{i=0}^{n-1} b_i p_i(x) = 0.$$

That is

$$\sum_{i=0}^{n-1} \frac{a_i}{x-\alpha^{-i}} \equiv 0 \pmod{x^{d-1}} \text{ and } \sum_{i=0}^{n-1} \frac{b_i}{x-\alpha^{-i}} \equiv 0 \pmod{x^{d-1}}.$$

This implies

$$\mathbf{c} = \mathbf{a} + \delta\mathbf{b} \in \mathbf{\Gamma}(L, x^{d-1})_{\mathbb{F}_q} + \delta\mathbf{\Gamma}(L, x^{d-1})_{\mathbb{F}_q} = \boldsymbol{R}S(n,k)_{\mathbb{F}_q} + \delta\boldsymbol{R}S(n,k)_{\mathbb{F}_q}$$

and

$$\mathbf{\Gamma}\left(L, x^{d-1}\right)_{\mathbb{F}_{q^2}} \subset \mathbf{\Gamma}(L, x^{d-1})_{\mathbb{F}_q} + \delta\mathbf{\Gamma}(L, x^{d-1})_{\mathbb{F}_q}.$$

We summarize this in the following proposition.

Proposition 3.3. *Let $\alpha \in \mathbb{F}_q$ be a primitive nth root of unity and $L = \{\alpha^0, \alpha^1, \cdots, \alpha^{n-1}\}$. Then,*

$$\mathbf{\Gamma}(L, x^{d-1})_{\mathbb{F}_{q^2}} = \mathbf{GU}(2,\ \mathbf{\Gamma}(L, x^{d-1})_{\mathbb{F}_q}).$$

a. Another way to see that it is as follows: If we take $d-1$ consecutive powers of α and $d-1 = n-k$, we know $\boldsymbol{R}S(n,k)$ has a parity check matrix given by

$$H = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \cdots & \alpha^{(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{d-2} & \alpha^{2(d-2)} & \cdots & \alpha^{(n-1)(d-2)} \\ 1 & \alpha^{d-1} & \alpha^{2(d-1)} & \cdots & \alpha^{(n-1)(d-1)} \end{pmatrix}.$$

For $\boldsymbol{c} = \boldsymbol{a} + \delta\boldsymbol{b} \in \mathbb{F}_{q^2}^n$, $\boldsymbol{c} \in \boldsymbol{R}S(n,k)_{\mathbb{F}_{q^2}}$ if and only if $H\boldsymbol{c}^t = 0$, i.e.,

$$\begin{aligned} 0 &= 1(a_1 + \delta b_1) + \alpha^j(a_2 + \delta b_2) + \cdots + \alpha^{j(n-1)}(a_n + \delta b_n) \\ &= \left(1a_1 + \alpha^j a_2 + \cdots + \alpha^{j(n-1)}a_n\right) + \delta\left(1b_1 + \alpha^j b_2 + \cdots + \alpha^{j(n-1)}b_n\right) \\ &\Rightarrow\ Ha^t + \delta Hb^t = 0 \\ &\Rightarrow\ Ha^t = 0 \ \textit{and}\ \ Hb^t = 0, \end{aligned}$$

because $1a_1 + \alpha^j a_2 + \cdots + \alpha^{j(n-1)}a_n \in \mathbb{F}_q$ and $1b_1 + \alpha^j b_2 + \cdots + \alpha^{j(n-1)}b_n \in \mathbb{F}_q$. Then, $\boldsymbol{R}S(n,k)_{\mathbb{F}_{q^2}} \subset \boldsymbol{R}S(n,k)_{\mathbb{F}_q} + \delta\boldsymbol{R}S(n,k)_{\mathbb{F}_q}$ and we conclude that

$$\boldsymbol{R}S(n,k)_{\mathbb{F}_{q^2}} = \mathbf{GU}(2,\ \boldsymbol{R}S(n,k)_{\mathbb{F}_q})$$

as the above proposition says.

In general, we have:

Theorem 3.3. *Let m be an even number, $L = \{\alpha_1, \cdots, \alpha_n\} \subset \mathbb{F}_{q^m}$ be such that $\alpha_i \in \mathbb{F}_q$, $1 \leq i \leq n$, and $g(x) = g_0 + g_1 x + \cdots + c_t x^t \in \mathbb{F}_{q^m}[x]$ be such that $g_j \in \mathbb{F}_q$, $1 \leq j \leq t$. Then,*

$$\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}} = \mathbf{GU}(2,\ \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q}).$$

Proof. Given $\boldsymbol{c} = \boldsymbol{a} + \delta\boldsymbol{b} \in \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q} + \delta\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q}$,

$$\sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} = \sum_{i=1}^{n} \frac{a_i + \delta b_i}{x - \alpha_i} = \sum_{i=1}^{n} \frac{a_i}{x - \alpha_i} + \delta \sum_{i=1}^{n} \frac{b_i}{x - \alpha_i} \equiv 0 \pmod{g(x)}$$

then

$$\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q} + \delta\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q} \subset \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}}.$$

On the other hand, we observe that $\mathbb{F}_{q^2}$ is a subfield of $\mathbb{F}_{q^m}$ because m is an even number. We know $p_i(x) = -\frac{g(x)-g(\alpha_i)}{x-\alpha_i} g^{-1}(\alpha_i) = a_{0i} + a_{1i}x + \cdots a_{t-1i}x^{t-1} \in \mathbb{F}_q[x]$ and a parity check matrix for $\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}}$ is given by

$$H = \begin{pmatrix} a_{01} & a_{02} & a_{03} & \cdots & a_{0n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{t-11} & a_{t-12} & a_{t-13} & \cdots & a_{t-1n} \end{pmatrix}.$$

Given $\boldsymbol{c} = \boldsymbol{a} + \delta\boldsymbol{b} \in \mathbb{F}_{q^2}^n$, $\boldsymbol{c} \in \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}}$ if and only if $H\boldsymbol{c}^t = 0$, i.e., for all $0 \leq j \leq t-1$,

$$\begin{aligned} 0 &= a_{j1}(a_1 + \delta b_1) + a_{j2}(a_2 + \delta b_2) + \cdots + a_{jn}(a_n + \delta b_n) \\ &= (a_{j1}a_1 + a_{j2}a_2 + \cdots + a_{jn}a_n) + \delta\,(a_{j1}b_1 + a_{j2}b_2 + \cdots + a_{jn}b_n) \\ &\Rightarrow\ Ha^t + \delta Hb^t = 0 \\ &\Rightarrow\ Ha^t = 0 \ \textit{and}\ \ Hb^t = 0, \end{aligned}$$

since $a_{j1}a_1 + a_{j2}a_2 + \cdots + a_{jn}a_n \in \mathbb{F}_q$ and $a_{j1}b_1 + a_{j2}b_2 + \cdots + a_{jn}b_n \ \in \mathbb{F}_q$.

From this $\boldsymbol{a},\ \boldsymbol{b} \in \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q}$, i.e., $\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}} \subset \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q} + \delta\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q}$. Thus

$$\mathbf{\Gamma}\left(L, g(x)\right)_{\mathbb{F}_{q^2}} = \mathbf{GU}(2,\ \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q}).$$

□

Remark: Since $\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}}$ is self-amalgamated, then it is Frobenius invariant. From *Theorem* 3.1 we obtain the other direction

Proposition 3.4. *Let m be an even number, $L = \{\alpha_1, \cdots, \alpha_n\} \subset \mathbb{F}_{q^m}$ and $g(x) = g_0 + g_1x + \cdots + c_tx^t \in \mathbb{F}_{q^m}[x]$ be such that for any codeword $\boldsymbol{c} \in \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}}$, $\boldsymbol{c}^q$ is also a codeword. Then,*

$$\mathbf{\Gamma}(L, g(x))_{\mathbb{F}_{q^2}} = \mathbf{GU}(2,\ \mathbf{\Gamma}(L, g(x))_{\mathbb{F}_q}).$$

3.3 DUAL OF THE AMALGAMATED CODE

We remember that for a linear code $C \subset \mathbb{F}_q^n$, the set $C^\perp = \left\{\boldsymbol{x} \in \mathbb{F}_q^n : \boldsymbol{x} \cdot \boldsymbol{y} = 0 \ \ \forall \boldsymbol{y} \in C\right\}$ is the dual of C with respect to the Euclidean inner product. We know $C^\perp$ is linear over $\mathbb{F}_q$ and

$$dim(C) + dim(C^\perp) = n.$$

That is, $C^\perp$ is a $[n,\ n-k]$ linear code, where $k = dim(C)$. The linear code C is called self-dual if $C^\perp = C$, and it is called self-orthogonal or weakly self-dual if $C \subseteq C^\perp$. We observe that $C^\perp$ is exactly the set of all parity checks on C. If $\beta = \{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_k\}$ is a basis of C over $\mathbb{F}_q$ and $G = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_k]$ is a generator matrix of C, then

$$C^\perp = \left\{\boldsymbol{b} \in \mathbb{F}_q^n : G\boldsymbol{b}^t = 0\right\}.$$

That is, G is a parity check matrix of $C^\perp$.

Now, suppose that C_0 is an $[n,\ k_0,\ d_0]$ linear code over $\mathbb{F}_q$. We know by Lemma 3.1 that $C = \mathbf{GU}(m,\ C_0)$ is an $[n,\ k_0,\ d_0]$ linear code over $\mathbb{F}_{q^m}$. We want to study the duality of the linear code $\mathbf{GU}(m,\ C_0)$.

Proposition 3.5. *Let $C_0 \subset \mathbb{F}_q^n$ be a linear code and m be an even number. Then,*

$$\mathbf{GU}(m,\ C_0)^{\perp} = \mathbf{GU}(m,\ C_0^{\perp}) = \mathbf{GU}(m,\ C_0)^{\perp_{TH}}$$

and

$$C_0 \subset C_0^{\perp} \rightarrow \ \ \mathbf{GU}(m,\ C_0) \subset \mathbf{GU}(m,\ C_0)^{\perp}$$

$$C_0 = C_0^{\perp} \rightarrow \ \ \mathbf{GU}(m,\ C_0) = \mathbf{GU}(m,\ C_0)^{\perp}.$$

Proof. Given $\boldsymbol{y} \in \mathbf{GU}(m,\ C_0^{\perp})$, we write $\boldsymbol{y} = \sum_{i=0}^{m-1} \delta^i \boldsymbol{b}_i$ where $\boldsymbol{b}_i \in C_0^{\perp}$. Writing $\boldsymbol{x} = \sum_{i=0}^{m-1} \delta^i \boldsymbol{a}_i \in \mathbf{GU}(m,\ C_0)$, and since $x_i = \sum_{j=0}^{m-1} \delta^j a_{ji}$ and $y_i = \sum_{k=0}^{m-1} \delta^k b_{ki}$, from Equation (3.7) we get

$$\begin{aligned}
\boldsymbol{x} \cdot \boldsymbol{y} &= \sum_{i=1}^{n} x_i y_i = \left(\boldsymbol{a}_0 + \delta \boldsymbol{a}_1 + \cdots + \delta^{m-1} \boldsymbol{a}_{m-1}\right) \cdot \left(\boldsymbol{b}_0 + \delta \boldsymbol{b}_1 + \cdots + \delta^{m-1} \boldsymbol{b}_{m-1}\right) \\
&= \left(\sum_{k=0}^{m-1} \delta^k \boldsymbol{a}_0 \cdot \boldsymbol{b}_k\right) + \delta \left(\sum_{k=0}^{m-1} \delta^k \boldsymbol{a}_1 \cdot \boldsymbol{b}_k\right) + \cdots + \delta^{m-1} \left(\sum_{k=0}^{m-1} \delta^k \boldsymbol{a}_{m-1} \cdot \boldsymbol{b}_k\right) \\
&= 0
\end{aligned}$$

That is, $\mathbf{GU}(m,\ C_0^{\perp}) \subset \mathbf{GU}(m,\ C_0)^{\perp}$. But $|\mathbf{GU}(m,\ C_0^{\perp})| = q^{n-k_0} q^{n-k_0} \cdots q^{n-k_0} = (q^m)^{n-k_0} = |\mathbf{GU}(m,\ C_0)^{\perp}|$, i.e., $\mathbf{GU}(m,\ C_0)^{\perp} = \mathbf{GU}(m,\ C_0^{\perp})$. Observe that if $C_0 \subset C_0^{\perp}$, $\mathbf{GU}(m,\ C_0) \subset \mathbf{GU}(m,\ C_0)^{\perp}$.

On the other hand, we want to find the Hermitian dual of the code $C \subset \mathbb{F}_{q^m}^n$. In this case $\overline{\boldsymbol{y}} = \boldsymbol{y}^{\sqrt{q^m}}$, that is, we take m to be an even number and the Frobenius automorphism with fixed field $\mathbb{F}_{q^{m/2}}$. Then, $\overline{\boldsymbol{y}} = (\boldsymbol{b}_0 + \overline{\delta} \boldsymbol{b}_1 + \cdots + \overline{\delta^{(m-1)}} \boldsymbol{b}_{m-1})$, because $\boldsymbol{b}_j \in \mathbb{F}_q^n$ and we get

$$\boldsymbol{x} \cdot_H \boldsymbol{y} := \boldsymbol{x} \cdot \overline{\boldsymbol{y}} = (\boldsymbol{a}_0 + \delta \boldsymbol{a}_1 + \cdots + \delta^{m-1} \boldsymbol{a}_{m-1}) \cdot (\boldsymbol{b}_0 + \overline{\delta} \boldsymbol{b}_1 + \cdots + \overline{\delta^{(m-1)}} \boldsymbol{b}_{m-1}) = 0,$$

because each $\boldsymbol{a}_i \cdot \boldsymbol{b}_j = 0$ for $0 \leq i,\ j \leq m-1$. Therefore

$$\boldsymbol{x} \cdot_{TH} \boldsymbol{y} = tr(\boldsymbol{x} \cdot_H \boldsymbol{y}) = 0,$$

where

$$tr : \mathbb{F}_{q^m} \to \mathbb{F}_{q^{m/2}}, \quad x \to x + x^{q^{m/2}}$$

is the trace of $\mathbb{F}_{q^m}$ over $\mathbb{F}_{q^{m/2}}$. That is, $\mathbf{GU}(m,\ C_0)^{\perp} \subset \mathbf{GU}(m,\ C_0)^{\perp_{TH}}$ and they have the same size.

□

Theorem 3.4. *Let $\mathbb{F}_q$ be a finite field of characteristic two and consider two different linear codes over $\mathbb{F}_q$, C_0 and C_1, such that $C_0 \subset C_1$. Let $\delta \in F_{q^2} \setminus F_q$ be as defined in Equation (3.1). Then, the* **additive code** *$C = \mathbf{GU}(C_0, C_1^{\perp}) \subset \mathbb{F}_{q^2}^n$ is Trace Hermitian self-orthogonal.*

Proof. From Lemma 3.1, $C = \mathbf{GU}(C_0, C_1^{\perp})$ is an additive nonlinear code over $\mathbb{F}_{q^2}$. Given $\boldsymbol{y} = \boldsymbol{c} + \delta \boldsymbol{d} \in C_1 + \delta C_0^{\perp}$ and any $\boldsymbol{x} = \boldsymbol{a} + \delta \boldsymbol{b} \in C$, from Equation (3.5),

$$\boldsymbol{x} \cdot_{TH} \boldsymbol{y} = 2(\mathbf{a} \cdot \mathbf{c} + \alpha \mathbf{b} \cdot \mathbf{d}) + \beta(\mathbf{a} \cdot \mathbf{d} - \mathbf{b} \cdot \mathbf{c}) = \beta(\mathbf{a} \cdot \mathbf{d} - \mathbf{b} \cdot \mathbf{c}) = \beta(0) = 0.$$

Then

$$C_1 + \delta C_0^{\perp} \subset (C_0 + \delta C_1^{\perp})^{\perp_{TH}}$$

But

$$|(C_0 + \delta C_1^{\perp})^{\perp_{TH}}| = q^{2n-(k_0+n-k_1)} = q^{n-(k_0-k_1)},$$

and

$$|C_1 + \delta C_0^{\perp}| = q^{k_1} q^{n-k_0} = q^{n-(k_0-k_1)}.$$

That is, $C^{\perp_{TH}} = (C_0 + \delta C_1^{\perp})^{\perp_{TH}} = C_1 + \delta C_0^{\perp}$ and since $C_0 \subset C_1$, $C \subset C^{\perp_{TH}}$, i.e., C is Trace Hermitian self-orthogonal. □

Example 3.7. *Let C_0 be the repetition code with parameters $[2^m,\ 1,\ 2^m]$. Then $C_0^{\perp}$ has parameters $[2^m,\ 2^m-1,\ 2]$. Let $C_1 = \boldsymbol{R}(1,\ m)$ with $C_1^{\perp} = \boldsymbol{R}(m-2,\ m)$ which has parameters $[2^m,\ 2^m-m-1,\ 4]$. We have $C_0 \subset C_1$ and $C = C_0 + \delta C_1^{\perp}$ is an* **additive and Trace Hermitian self-orthogonal code over** $\mathbb{F}_4$ *with parameters $(2^m,\ 2^{2^m-m},\ 4)$. Observe that $C^{\perp_{TH}} = C_1 + \delta C_0^{\perp}$ has parameters $(2^m,\ 2^{2^m+m},\ 2)$.*

Example 3.8. *We take C_0 and C_1 as Example 3.7. Since $C_0 \subset C_1^{\perp}$, from Theorem 3.4, we let $C = C_0 + \delta C_1$, an additive code with parameters $(2^m,\ 2^{m+2},\ 2^{m-1})$. Then $C \subset C^{\perp_{TH}}$. Observe that for $\boldsymbol{c}+\delta\boldsymbol{d}$ in $C_1^{\perp} + \delta C_0^{\perp}$, $(\boldsymbol{a}+\delta\boldsymbol{b})\cdot(\boldsymbol{c}+\delta\boldsymbol{d}) = \boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c} = 0+0=0$ and*

$$|C_1^{\perp} + \delta C_0^{\perp}| = 2^{2^m-m-1}2^{2^m-1} = 2^{2^{m+1}-m-2} = |C^{\perp_{TH}}|.$$

That is $C^{\perp_{TH}} = C_1^{\perp} + \delta C_0^{\perp}$, which has parameters $(2^m,\ 2^{2^{m+1}-m-2},\ d)$.

Lemma 3.3. *Let $\mathbb{F}_q$ be a finite field of characteristic two and consider a linear code $C_0 \subset \mathbb{F}_q^n$. The* **additive code** $C = \mathbf{GU}(C_0, C_0^{\perp}) \in \mathbb{F}_{q^2}^n$ *is Trace Hermitian self-dual.*

Proof. From Lemma 3.1, if $C_0^{\perp} \neq C_0$, $C = C_0 + \delta C_0^{\perp}$ is an additive nonlinear code over $\mathbb{F}_{q^2}$. For $\boldsymbol{y} = \boldsymbol{c} + \delta\boldsymbol{d}$ and $\boldsymbol{x} = \boldsymbol{a} + \delta\boldsymbol{b}$ in $C_0 + \delta C_0^{\perp}$, from Equation (3.5),

$$\boldsymbol{x} \cdot_{TH} \boldsymbol{y} = 2(\mathbf{a}\cdot\mathbf{c} + \alpha\mathbf{b}\cdot\mathbf{d}) + \beta(\mathbf{a}\cdot\mathbf{d} - \mathbf{b}\cdot\mathbf{c}) = \beta(\mathbf{a}\cdot\mathbf{d} - \mathbf{b}\cdot\mathbf{c}) = \beta(0) = 0,$$

i.e., $C \subset C^{\perp_{TH}}$. But

$$|C^{\perp_{TH}}| = q^{2n-(k_0+n-k_0)} = q^n$$

and

$$|C| = |C_0 + \delta C_0^{\perp}| = q^{k_0}q^{n-k_0} = q^n.$$

That is, $C^{\perp_{TH}} = C$. □

Example 3.9. *Let $C_0 = S_k(2)$ be the binary simplex code of dimension k, this linear code has parameters $[2^k-1,\ k,\ 2^{k-1}]$. Now, $C_0^{\perp} = S_k(2)^{\perp} = H_k(2)$, the binary Hamming code with parameters $[2^k-1,\ 2^k-1-k,\ 3]$. Then the additive code*

over $\mathbb{F}_4$ *given by* $\mathbf{GU}(S_k(2), H_k(2))$ *is Trace Hermitian self-dual with parameters* $(2^k-1,\ 2^{2^k-1},\ 3)$, *see Example 3.2.*

Now, we take the particular case $q=2$ and $\delta \in \mathbb{F}_4$ a root of x^2+x+1, that is, $\alpha=1$ and $\beta=1$, see Equation (3.1). Then $tr(\delta)=\delta+\delta^2=1$ and the conjugate of δ is $\overline{\delta}=\delta^2=\delta+1$. Given $\boldsymbol{a}+\delta\boldsymbol{b}$ and $\boldsymbol{c}+\delta\boldsymbol{d}$ in $\mathbb{F}_4^n$, from Equation (3.2) to Equation (3.5) we obtain

a. Euclidean inner product

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot(\boldsymbol{c}+\delta\boldsymbol{d}) = (\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d}). \tag{3.11}$$

b. Trace Euclidean inner product

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_T(\boldsymbol{c}+\delta\boldsymbol{d}) = tr[(\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})] = \boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d}. \tag{3.12}$$

c. Hermitian inner product

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_H(\boldsymbol{c}+\delta\boldsymbol{d}) = (\boldsymbol{a}+\delta\boldsymbol{b})\cdot(\boldsymbol{c}+\delta^2\boldsymbol{d}) = (\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d}+\boldsymbol{a}\cdot\boldsymbol{d})+\delta(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}). \tag{3.13}$$

d. Trace Hermitian inner product

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_{TH}(\boldsymbol{c}+\delta\boldsymbol{d}) = tr[(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_H(\boldsymbol{c}+\delta\boldsymbol{d})] = \boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}. \tag{3.14}$$

We have to say, as in *Theorem* 3 of [14], that the following is true

Proposition 3.6. *Let* $C \subset \mathbb{F}_4^n$ *be a linear code,* $C^{\perp}$ *the dual of* C *with respect to Euclidean inner product and* $C^{\perp_T}$ *the dual of* C *with respect to the Trace Euclidean inner product. Then* C *is self-orthogonal with respect to one of them if and only if it is self-orthogonal with respect to the other.*

Proof. Let $\boldsymbol{x} \in C$ and $\boldsymbol{y} \in \mathbb{F}_4^n$. Now we can write $\boldsymbol{x}=\boldsymbol{a}+\delta\boldsymbol{b}$, for some $\boldsymbol{a},\ \boldsymbol{b} \in \mathbb{F}_2^n$ and $\boldsymbol{y}=\boldsymbol{c}+\delta\boldsymbol{d}$ for some $\boldsymbol{c},\ \boldsymbol{d} \in \mathbb{F}_2^n$. We know

$$\boldsymbol{x}\cdot\boldsymbol{y} = (\boldsymbol{a}+\delta\boldsymbol{b})\cdot(\boldsymbol{c}+\delta\boldsymbol{d}) = (\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})$$

and

$$\boldsymbol{x} \cdot_T \boldsymbol{y} = tr(\boldsymbol{x} \cdot \boldsymbol{y}) = tr((\boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d}) + \delta(\boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d})) = \boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d}.$$

If $\boldsymbol{y} \in C^{\perp}$, $\boldsymbol{x} \cdot \boldsymbol{y} = 0$ for all $\boldsymbol{x} \in C$, then $\boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d} = 0$ and $\boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d} = 0$. That is, $\boldsymbol{x} \cdot_T \boldsymbol{y} = 0$ for all $\boldsymbol{x} \in C$ and then $\boldsymbol{y} \in C^{\perp_T}$.
On the other hand, if $\boldsymbol{y} = \boldsymbol{c} + \delta \boldsymbol{d} \in C^{\perp_T}$, from Equation (3.3) and Equation (3.12),

$$\boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d} = 0.$$

Since C is linear, $C^{\perp_T}$ is linear over $\mathbb{F}_4$. We get that $\delta^2 \boldsymbol{y} \in C^{\perp_T}$, and

$$\boldsymbol{x} \cdot_T \delta^2 \boldsymbol{y} = tr(\delta^2 \boldsymbol{x} \cdot \boldsymbol{y}) = tr(\delta^2(\boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d}) + (\boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d})) = tr(\delta(\boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d}) + (\boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} + \boldsymbol{a} \cdot \boldsymbol{c})).$$

That is

$$\boldsymbol{x} \cdot_T \delta^2 \boldsymbol{y} = \boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d} = 0.$$

Since $\boldsymbol{x} \cdot \boldsymbol{y} = (\boldsymbol{a} + \delta \boldsymbol{b}) \cdot (\boldsymbol{c} + \delta \boldsymbol{d}) = (\boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d}) + \delta(\boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d})$, we get, $\boldsymbol{x} \cdot \boldsymbol{y} = 0$ for all $\boldsymbol{x} \in C$. That is, $\boldsymbol{y} \in C^{\perp}$ and we get $C^{\perp} = C^{\perp_T}$. □

Proposition 3.7. *Let $C_0 \subset \mathbb{F}_2^n$ be an $[n,\ k,\ d]$ linear code. If C_0 is self-orthogonal with respect to the Euclidean inner product, then $C = \mathbf{GU}(2,\ C_0) \subset \mathbb{F}_4^n$ is such that*

$$C \subset C^{\perp} = C^{\perp_{TH}},$$

$$C \subset C^{\perp} = C^{\perp_H},$$

$$C \subset C^{\perp} = C^{\perp_T}.$$

Proof. From *Proposition* 3.5, we obtain that $C^{\perp} = C^{\perp_{TH}} = C^{\perp_T} = C^{\perp_H}$ and, since C_0 is self-orthogonal, we get that C is also self-orthogonal.

□

3.4 AN INTERESTING SPECIAL CASE

We consider a prime $q \equiv 3 \pmod 4$, observe that the **Mersenne prime numbers are example of such integer** q**.**

Taking q in this particular form we may obtain a general version of the *Theorem* 3 of [14]: First, we observe the following known result that follows from quadratic reciprocity (see Chapter 5, Ireland and Rosen [29]).

a. For any prime q, such that $q \equiv 3 \pmod 4$, the polynomial $x^2 + 1$ is irreducible over $\mathbb{F}_q$.

b. In this case, the irreducible polynomial has the form $p(x) = x^2 + 1$, that is, $\alpha = 1$ and $\beta = 0$, see Equation (3.1). If $\delta \in \mathbb{F}_{q^2}$ is such that $\delta^2 + 1 = 0$, $\delta^2 = -1$, the set $\{1,\ \delta\}$ is a basis, polynomial basis, of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$ because the the matrix

$$A = \begin{pmatrix} 1 & \delta \\ 1 & \delta^q \end{pmatrix}$$

is such that $det(A) = \delta^q - \delta = -2\delta \neq 0$. If $-2\delta = 0$ then $2|q$ which is not possible. Observe that $\delta^q = (\delta^2)^m\delta = (-1)^m\delta = -\delta$ because m is odd. That is, the conjugate of δ over $\mathbb{F}_q$ is $-\delta$. Also, we get that the trace of δ is zero because

$$tr(\delta) = \delta + \delta^q = \delta - \delta = 0 = -\beta.$$

Now, for $\boldsymbol{a} + \delta\boldsymbol{b}$, $\boldsymbol{c} + \delta\boldsymbol{d}$ in $\mathbb{F}_{q^2}^n$, since in this case $\alpha = 1$ and $\beta = 0$, from Equation (3.2) to Equation (3.5) we get:

a. Euclidean inner product

$$(\boldsymbol{a} + \delta\boldsymbol{b}) \cdot (\boldsymbol{c} + \delta\boldsymbol{d}) = (\boldsymbol{a} \cdot \boldsymbol{c} - \boldsymbol{b} \cdot \boldsymbol{d}) + \delta(\boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c}). \tag{3.15}$$

b. Trace Euclidean inner product

$$(\boldsymbol{a}+\delta\boldsymbol{b}) \cdot_T (\boldsymbol{c}+\delta\boldsymbol{d}) = tr[(\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c})] = 2(\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d}). \tag{3.16}$$

c. Hermitian inner product

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_H(\boldsymbol{c}+\delta\boldsymbol{d}) = (\boldsymbol{a}+\delta\boldsymbol{b})\cdot(\boldsymbol{c}-\delta\boldsymbol{d}) = (\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{b}\cdot\boldsymbol{c}-\boldsymbol{a}\cdot\boldsymbol{d}). \quad (3.17)$$

d. Trace Hermitian inner product

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_{TH}(\boldsymbol{c}+\delta\boldsymbol{d}) = tr[(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_H(\boldsymbol{c}+\delta\boldsymbol{d})] = 2(\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d}). \quad (3.18)$$

Proposition 3.8. *Take q a prime such that $q = 2m+1$ with m odd. Let $C \subset \mathbb{F}_{q^2}^n$ be a linear code, $C^{\perp}$ the dual of C with respect to Euclidean inner product, and $C^{\perp_T}$ the dual of C with respect to the Trace Euclidean inner product. Then, C is self-orthogonal with respect to one of them if and only if it is self-orthogonal with respect to the other.*

Proof. Let $\boldsymbol{y} = \boldsymbol{c}+\delta\boldsymbol{d} \in C^{\perp}$, for all $\boldsymbol{a}+\delta\boldsymbol{b} \in C$, from Equation (3.15),

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot(\boldsymbol{c}+\delta\boldsymbol{d}) = (\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}) = 0.$$

Then, $\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d} = 0$ and $2(\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d}) = 0$. That is,

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_T(\boldsymbol{c}+\delta\boldsymbol{d}) = 0.$$

Thus, $\boldsymbol{y} \in C^{\perp_T}$ and

$$C^{\perp} \subset C^{\perp_T}.$$

On the other hand, if $\boldsymbol{y} = \boldsymbol{c}+\delta\boldsymbol{d} \in C^{\perp_T}$, for any $\boldsymbol{a}+\delta\boldsymbol{b} \in C$, and from Equation (3.16),

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_T(\boldsymbol{c}+\delta\boldsymbol{d}) = 2(\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d}) = 0.$$

That is, $\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d} = 0$. Since $C^{\perp_T}$ is linear over $\mathbb{F}_{q^2}$, $\delta\boldsymbol{y} \in C^{\perp_T}$ and

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_T(\delta\,\boldsymbol{y}) = (\boldsymbol{a}+\delta\boldsymbol{b})\cdot_T(\delta\,\boldsymbol{c}-\boldsymbol{d}) = -2(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}) = 0,$$

i.e, $\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c}=0$. Then,

$$0=(\boldsymbol{a}\cdot\boldsymbol{c}-\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{a}\cdot\boldsymbol{d}+\boldsymbol{b}\cdot\boldsymbol{c})=(\boldsymbol{a}+\delta\boldsymbol{b})\cdot(\boldsymbol{c}+\delta\boldsymbol{d}).$$

and we obtain that $\boldsymbol{y}\in C^{\perp}$. Thus

$$C^{\perp}=C^{\perp_T}.$$

□

The following result is a general version of the *Theorem* 3 of [14].

Theorem 3.5. *Let q be a prime such that $q\equiv 3 \pmod 4$, $C\subset\mathbb{F}_{q^2}^n$ be a linear code, $C^{\perp_H}$ the dual of C with respect to Hermitian inner product, and $C^{\perp_{TH}}$ the dual of C with respect to the Trace Hermitian inner product. Then, C is self-orthogonal with respect to one of them if and only if it is self-orthogonal with respect to the other.*

Proof. Let $\boldsymbol{y}=\boldsymbol{c}+\delta\boldsymbol{d}\in C^{\perp_H}$, for all $\boldsymbol{a}+\delta\boldsymbol{b}\in C$, from equation (3.17),

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_H(\boldsymbol{c}+\delta\boldsymbol{d})=(\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})+\delta(\boldsymbol{b}\cdot\boldsymbol{c}-\boldsymbol{a}\cdot\boldsymbol{d})=0.$$

Then, $2(\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d})=0$. That is,

$$(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_{TH}(\boldsymbol{c}+\delta\boldsymbol{d})=0,$$

and we get that

$$C^{\perp_H}\subset C^{\perp_{TH}}.$$

On the other hand, if

$$\boldsymbol{y}=\boldsymbol{c}+\delta\boldsymbol{d}\in C^{\perp_{TH}}$$

for any $\boldsymbol{a}+\delta\boldsymbol{b}\in C$, from Equation (3.18) we obtain

$$0=(\boldsymbol{a}+\delta\boldsymbol{b})\cdot_{TH}(\boldsymbol{c}+\delta\boldsymbol{d})=2(\boldsymbol{a}\cdot\boldsymbol{c}+\boldsymbol{b}\cdot\boldsymbol{d}).$$

That is, $\boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d} = 0$. Now, $\delta\ \boldsymbol{y} = \delta\ \boldsymbol{c} - \boldsymbol{d} \in C^{\perp_{TH}}$ because C is linear and therefore $C^{\perp_{TH}}$ is also linear. Then,

$$0 = (\boldsymbol{a} + \delta\boldsymbol{b}) \cdot_{TH} (\delta\ \boldsymbol{y}) = tr[(\boldsymbol{a} + \delta\boldsymbol{b}) \cdot (-\boldsymbol{d} - \delta\boldsymbol{c})] = 2(\boldsymbol{b} \cdot \boldsymbol{c} - \boldsymbol{a} \cdot \boldsymbol{d}),$$

i.e., $\boldsymbol{b} \cdot \boldsymbol{c} - \boldsymbol{a} \cdot \boldsymbol{d} = 0$. Thus, $(\boldsymbol{a} + \delta\boldsymbol{b}) \cdot_H (\boldsymbol{c} + \delta\boldsymbol{d}) = (\boldsymbol{a} \cdot \boldsymbol{c} + \boldsymbol{b} \cdot \boldsymbol{d}) + \delta(\boldsymbol{b} \cdot \boldsymbol{c} - \boldsymbol{a} \cdot \boldsymbol{d}) = 0$, and we get

$$C^{\perp_H} = C^{\perp_{TH}}.$$

□

3.5 APPLICATIONS

Linear codes with few weights have applications in cryptography, association schemes, designs, strongly regular graphs, finite group theory, finite geometries, among other disciplines. For a comprehensive survey of two-weight codes, see [16], and for three and few-weights codes see [17, 21, 22]. We use our **GU** code construction to obtain two-weight, three-weight, and few-weights codes. Consequently, we also give an elementary construction of the two-weight codes in Calderbank and Kantor [16], of three-weight codes and few-weights codes given by Ding [22], and by Tonchev and Jungnickel [17].

3.5.1 TWO-WEIGHT CODES FROM THE GO-UP CONSTRUCTION

Let $C_0 \subset \mathbb{F}_2^n$ be an $[n,\ k,\ d]_2$ linear code such that for all $\boldsymbol{a} \neq 0 \in C_0$, $\omega(\boldsymbol{a}) = w_1$. That is, C_0 has the following weight distribution: $A_0 = 1$ and $A_{w_1} = 2^k - 1$, i.e., C_0 is a **one-weight code**. The binary **simplex code** with parameters $[2^k - 1,\ k,\ 2^{k-1}]_2$ is an example of a code C_0, see Proposition 2.1.

We want to calculate $\hat{A}_i = |\{\boldsymbol{x} \in \mathbf{GU}(2,\ C_0) : \omega(\boldsymbol{x}) = i\}|$. We know $\hat{A}_0 = 1$. Now, calculating $\hat{A}_{w_1} = |\{\boldsymbol{a} + \delta\boldsymbol{b} \in C \mid \omega(\boldsymbol{a} + \delta\boldsymbol{b}) = w_1\}|$ we observe that for $\boldsymbol{a} \neq 0$ the codewords of the form $\boldsymbol{a} + \delta\mathbf{0} \in C$ have weight w_1, same for $\mathbf{0} + \delta\boldsymbol{b}$ with $\boldsymbol{b} \neq 0$. If $\boldsymbol{a} \neq \mathbf{0}$, the codewords $\boldsymbol{a} + \delta\boldsymbol{a}$ also have weight w_1. Therefore,

$$\hat{A}_{w_1} = 3A_{w_1} = 3(2^k - 1).$$

The other codewords have weight $\frac{3w_1}{2}$, because, if $\boldsymbol{a} \neq \boldsymbol{b}$ and both are different to zero, then

$$\omega(\boldsymbol{a} + \boldsymbol{b}) = w_1 = \omega(\boldsymbol{a}) + \omega(\boldsymbol{b}) - 2\omega(\boldsymbol{a} * \boldsymbol{b}) = 2w_1 - 2\omega(\boldsymbol{a} * \boldsymbol{b}),$$

i.e., $\omega(\boldsymbol{a} * \boldsymbol{b}) = \frac{w_1}{2}$. Thus,

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = \omega(\boldsymbol{a}) + \omega(\boldsymbol{b}) - \omega(\boldsymbol{a} * \boldsymbol{b}) = 2w_1 - \frac{w_1}{2} = \frac{3w_1}{2}$$

for all $\boldsymbol{a} \neq 0$, $\boldsymbol{b} \neq 0$ and $\boldsymbol{a} \neq \boldsymbol{b}$. Then

$$\begin{aligned}
\hat{A}_{\frac{3w_1}{2}} &= 2^{2k} - 3A_{w_1} - 1 \\
&= 2^{2k} - 3(2^k - 1) - 1 \\
&= 2^{2k} - 2^k - 2(2^k) + 2 \\
&= 2^k(2^k - 1) - 2(2^k - 1) \\
&= A_{w_1}(A_{w_1} - 1).
\end{aligned}$$

Remark: Since $\omega(\boldsymbol{a} * \boldsymbol{b})$ is an integer and we have obtained $\omega(\boldsymbol{a} * \boldsymbol{b}) = \frac{w_1}{2}$, then w_1 is even. When C_0 is the binary simplex code with $k \geq 3$, $\mathbf{GU}(2,\ C_0)$ is Euclidean self-orthogonal and $\mathbf{GU}(2,\ C_0)^{\perp} = C_0^{\perp} + \delta C_0^{\perp}$.

Table 3–8 gives the weight distribution of $\mathbf{GU}(2,\ C_0)$.

i	$\hat{A}_i$
0	1
w_1	$3A_{w_1}$
$\frac{3w_1}{2}$	$A_{w_1}(A_{w_1} - 1)$

Table 3–8: *Weight Distribution of a Two-Weight Code over* $\mathbb{F}_4$

Let $C_0 \subset \mathbb{F}_q^n$ be a **one-weight** code, i.e., for $\boldsymbol{a} \in C_0 \setminus \{0\}$, $\omega(\boldsymbol{a}) = w_1$ and $A_{w_1} = |\{\boldsymbol{a} \in C_0 \ :\ \omega(\boldsymbol{a}) = w_1\}|$

Definition 3.3. *Given $C_0 \subset \mathbb{F}_q^n$ a linear code. We say that C_0 has the property of* **constant intersection** *if for any $\boldsymbol{a}$, $\boldsymbol{b}$ in $C_0 \setminus \{0\}$ with $\boldsymbol{b} \neq \lambda\boldsymbol{a}$ and $\lambda \in \mathbb{F}_q^*$, we have $\omega(\boldsymbol{a} * \boldsymbol{b}) = I$ is a constant number, where $\boldsymbol{a} * \boldsymbol{b} = (c_1\ \cdots\ c_n)$ and $c_i = a_i b_i$ if and only if $a_i + b_i = 0$ with $a_i \neq 0$ and $b_i \neq 0$. Otherwise $c_i = 0$.*

For example, when $q = 2$ and C_0 is the binary simplex code, $I = \frac{w_1}{2}$. In general, we have

Lemma 3.4. *The simplex code $S_k(q)$ with parameters $[\frac{q^k-1}{q-1},\ k,\ q^{k-1}]$ has the property of constant intersection with $I = \frac{w_1}{q} = q^{k-2}$.*

Proof. If $\boldsymbol{a}$ and $\boldsymbol{b}$ in $S_k(q)$ are different nonzero elements with $\boldsymbol{b} = \lambda\boldsymbol{a}$ for some $\lambda \in \mathbb{F}_q^*$ such that $1 + \lambda \neq 0$, then $\boldsymbol{a} + \boldsymbol{b} = (1 + \lambda)\boldsymbol{a} \neq 0$ and

$$\omega(\boldsymbol{a} + \boldsymbol{b}) = \omega(\boldsymbol{a}) = q^{k-1}.$$

From this

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = \omega(\boldsymbol{a}) = q^{k-1}.$$

If $1 + \lambda = 0$, $\boldsymbol{a} + \boldsymbol{b} = 0$, but, since

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = \omega(\boldsymbol{a} + \boldsymbol{b}) + \omega(\boldsymbol{a} * \boldsymbol{b}), \tag{3.19}$$

we get that

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = 0 + q^{k-1} = q^{k-1}.$$

Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be nonzero elements of $S_k(q)$ with $\boldsymbol{b} \neq \lambda\boldsymbol{a}$, $\lambda \in \mathbb{F}_q^*$. Let $a_j \neq 0$, $b_j \neq 0$ be such that $a_j + b_j = 0$, $\lambda a_j + \lambda b_j = 0$ for any $\lambda \in \mathbb{F}_q^*$. Then, there exists $P_j = (x_{1j}\ \cdots\ x_{kj}) \in PG(k-1, q)$ where $a_j = x_{rj}$, $b_j = x_{sj}$. Since for any $\lambda \in \mathbb{F}_q^*$ $\lambda\ P_j$ generates the same one-dimensional subspace, we may take either $a_j = 1$ and $b_j = q - 1$ or $a_j = q - 1$ and $b_j = 1$. If either $a_j = 1$ and $b_j = q - 1$ or $a_j = q - 1$ and $b_j = 1$, the other $k - 2$ entries of P_j can take any value from $\mathbb{F}_q$, i.e., we have q^{k-2} possible value for the other entries. We have each P_j, taken as a representative of the one-dimensional subspace of $\mathbb{F}_q^k$, appears one time as a column in the standard generator matrix of $S_k(q)$, we get q^{k-2} points of the projective geometry such that $x_{rj} + x_{sj} = 0$. Thus,

$$\omega(\boldsymbol{a} * \boldsymbol{b}) = q^{k-2}.$$

□

Theorem 3.6. *Let* $C_0 \subset \mathbb{F}_q^n$ *be a* **one-weight** *linear code with the property of* **constant intersection.** *Then,* $\mathbf{GU}(2,\ C_0)$ *is a linear* **two-weight** *code over* $\mathbb{F}_{q^2}$ *and its weight distribution is given by* $\hat{A}_{w_1} = (q+1)A_{w_1}$ *and* $\hat{A}_{w_2} = A_{w_1}(A_{w_1} - q + 1)$, *where* $A_{w_1} = |\{\boldsymbol{a} \in C_0 \ :\ \omega(\boldsymbol{a}) = w_1\}| = q^k - 1$.

Proof. Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be linearly independent over $\mathbb{F}_q$, i.e., $\boldsymbol{b} \notin \{\alpha\boldsymbol{a} \mid \alpha \in \mathbb{F}_q\}$. Then, from Equation (3.19),

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = \omega(\boldsymbol{a} + \boldsymbol{b}) + \omega(\boldsymbol{a} * \boldsymbol{b}) = w_1 + I = w_2.$$

In the case of $S_k(q)$, from *Lemma* 3.4,

$$w_2 = q^{k-1} + q^{k-2} = q^{k-2}(q+1) = \frac{q+1}{q} w_1$$

for all $q \geq 2$. We get $(q^k - 1)(q^k - q)$ linearly independent sets of the form $\{\boldsymbol{a}, \boldsymbol{b}\}$, that is,

$$\hat{A}_{w_2} = (q^k - 1)(q^k - q) = A_{w_1}(A_{w_1} - q + 1).$$

If the set $\{\boldsymbol{a}, \boldsymbol{b}\}$ is linearly dependent over $\mathbb{F}_q$, $\boldsymbol{b} \in \{\alpha\boldsymbol{a} \mid \alpha \in \mathbb{F}_q\}$, then $\omega(\boldsymbol{a} + \delta\alpha\boldsymbol{a}) = w_1$. From this, $\mathbf{GU}(2,\ C_0)$ is a **two-weight** linear code with

$$\begin{aligned}
\hat{A}_{w_1} &= q^{2k} - A_{w_2} - 1 \\
&= q^{2k} - (q^k - 1)(q^k - q) - 1 \\
&= q^{k+1} + q^k - q - 1 \\
&= q^k(q+1) - (q+1) \\
&= (q+1)(q^k - 1) = (q+1)A_{w_1}.
\end{aligned}$$

□

3.5.2 AN ELEMENTARY CONSTRUCTION OF A CLASS OF TWO-WEIGHT CODES WITH PARAMETERS OF (RT1) OF CALDERBANK AND KANTOR

As a consequence of Theorem 3.6, we derive the following infinite class of two-weight linear codes that have the same parameters as the codes given by Calderbank and Kantor [16]. To construct their class of two-weight codes RT1, they use relatively deep results from finite geometries. We derive our class as an immediate corollary.

Corollary 3.1. *There exists a two-weight code over* $\mathbb{F}_{q^2}$ *with parameters* $[\frac{q^k-1}{q-1}, k, q^{k-1}, (q+1)q^{k-2}]$

Proof. Take $C_0 = S_k(q)$ in Theorem 3.6. □

As stated, the class of two-weight codes with the parameters of Corollary 3.1 are also obtained by Calderbank and Kantor in [16] from a construction they called **RT**1. That construction uses some deep properties of finite projective geometry and rank 3− permutation groups. We get the same parameters with the amalgamation of the simplex code; our construction is much more elementary. It is possible that the two-weight codes with these parameters are unique. In that case, the codes we obtain can be equivalent to those obtained in [16].

Example 3.10. *Let* $C_0 = S_k(2) \subset \mathbb{F}_2^n$ *be the binary* $k-$*dimensional simplex code which is a code with parameters* $[2^k - 1,\ k,\ 2^{k-1}]$. *We obtain, from Theorem 3.6, Table 3–9 of two-weight codes over* $\mathbb{F}_4$.

k	$[2^k-1,\ k,\ 2^{k-1}]_4$	*Weight Distribution*		
2	$[3,\ 2,\ 2]_4$	$\hat{A}_0 = 1$	$\hat{A}_2 = 9$	$\hat{A}_3 = 6$
3	$[7,\ 3,\ 4]_4$	$\hat{A}_0 = 1$	$\hat{A}_4 = 21$	$\hat{A}_6 = 42$
4	$[15,\ 4,\ 8]_4$	$\hat{A}_0 = 1$	$\hat{A}_8 = 45$	$\hat{A}_{12} = 210$
5	$[31,\ 5,\ 16]_4$	$\hat{A}_0 = 1$	$\hat{A}_{16} = 93$	$\hat{A}_{24} = 930$
6	$[63,\ 6,\ 32]_4$	$\hat{A}_0 = 1$	$\hat{A}_{32} = 189$	$\hat{A}_{48} = 3906$
7	$[127,\ 7,\ 64]_4$	$\hat{A}_0 = 1$	$\hat{A}_{64} = 381$	$\hat{A}_{96} = 16002$
8	$[255,\ 8,\ 128]_4$	$\hat{A}_0 = 1$	$\hat{A}_{128} = 765$	$\hat{A}_{192} = 64770$
9	$[511,\ 9,\ 256]_4$	$\hat{A}_0 = 1$	$\hat{A}_{256} = 1533$	$\hat{A}_{384} = 260610$
10	$[1023,\ 10,\ 512]_4$	$\hat{A}_0 = 1$	$\hat{A}_{512} = 3069$	$\hat{A}_{768} = 1045506$

Table 3–9: *Examples of two-weight codes over* $\mathbb{F}_4$

Example 3.11. *Let* $C_0 = S_2(3)$ *be the simplex code with parameters* $[4,\ 2,\ 3]$. *A generator matrix is given by*

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix}.$$

Then $S_2(3) = \{\alpha(1\ 0\ 1\ 1) + \lambda(0\ 1\ 1\ 2) \mid \alpha, \lambda \in \mathbb{F}_3\}$ *and its elements are shown in Table 3–6.*

We get that, for $\boldsymbol{a}, \boldsymbol{b} \in S_2(3)$, $\omega(\boldsymbol{a} + \delta\boldsymbol{b}) \in \{3, 4\}$, $\hat{A}_3 = 32$ *and* $\hat{A}_4 = 48$.

Proposition 3.9. *If* $C_0 \subset \mathbb{F}_q^n$ *is a projective code, then* $\mathbf{GU}(2,\ C_0)$ *is a projective code.*

Proof. Let G be a generator matrix of C_0. Then no two columns of G are $\mathbb{F}_q-$dependent. From Proposition 3.1 any two columns of G are $\mathbb{F}_{q^2}-$independent. Then $\mathbf{GU}(2,\ C_0)$ is also a projective code. □

Remark: If C_0 is a constant weight and projective code, from Theorems 2.8 and 3.6 we get a projective $(n, k, n - w_1, n - \frac{q+1}{q}w_1)$ set. If we take $C_0 = S_k(q)$, we

get a projective set with parameters

$$\left(\frac{q^k-1}{q-1},\ k,\ \frac{q^{k-1}-1}{q-1},\ \frac{q^{k-2}-1}{q-1}\right).$$

3.5.3 GO-UP OF A TWO-WEIGHT CODES

Let $C_0 \subset \mathbb{F}_2^n$ be an $[n,\ k,\ d]$ linear code such that for all $\boldsymbol{a} \neq 0 \in C_0$, either $\omega(\boldsymbol{a}) = w_1$ or $\omega(\boldsymbol{a}) = w_2$. We take

$$C_{w_i} = \{\boldsymbol{a} \in C_0 \mid \omega(\boldsymbol{a}) = w_i\},$$

where $i = 1,\ 2$. Observe that

$$C_0 = \{0\} \cup C_{w_1} \cup C_{w_2}.$$

We suppose that w_2 and w_1 are even numbers with $w_1 < w_2$. Then $w_2 = w_1 + s$ for some even number s.

a. For $\boldsymbol{a} \in C_{w_1}$, $\boldsymbol{a}+\delta\boldsymbol{0}$, $\boldsymbol{0}+\delta\boldsymbol{a}$ and $\boldsymbol{a}+\delta\boldsymbol{a}$ are elements of $\mathbf{GU}(2,\ C_0)$ with Hamming weight w_1. Then $\hat{A}_{w_1} \geq 3A_{w_1}$. But if $\boldsymbol{a} \in C_{w_1}$ and $\boldsymbol{b} \in C_{w_2}$, $\omega(\boldsymbol{a}+\delta\boldsymbol{b}) \geq w_2 \neq w_1$. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are elements of C_{w_2}, $\omega(\boldsymbol{a}+\delta\boldsymbol{b}) \geq w_2 \neq w_1$. Thus,

$$\hat{A}_{w_1} = 3A_{w_1}.$$

b. For $\boldsymbol{a} \in C_{w_2}$, $\boldsymbol{a}+\delta\boldsymbol{0}$, $\boldsymbol{0}+\delta\boldsymbol{a}$ and $\boldsymbol{a}+\delta\boldsymbol{a}$ are elements of $\mathbf{GU}(2,\ C_0)$ with Hamming weight w_2. Then $\hat{A}_{w_2} \geq 3A_{w_2}$.

c. Let $\boldsymbol{a} \in C_{w_1}$ and $\boldsymbol{b} \in C_{w_1}$ with $\boldsymbol{a} \neq \boldsymbol{b}$. Since $\boldsymbol{a}+\boldsymbol{b} \in C_0$, either $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_1$ or $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_2$. In the first case,

$$w_1 = \omega(\boldsymbol{a}+\boldsymbol{b}) = w_1 + w_1 - 2\omega(\boldsymbol{a} * \boldsymbol{b}).$$

That is, $\omega(\boldsymbol{a} * \boldsymbol{b}) = \frac{w_1}{2}$. Then,

$$\omega(\boldsymbol{a}+\delta\boldsymbol{b}) = 2w_1 - \frac{w_1}{2} = \frac{3w_1}{2}.$$

If $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_2$, then

$$w_2 = 2w_1 - 2\omega(\boldsymbol{a} * \boldsymbol{b}),$$

i.e., $2\omega(\boldsymbol{a} * \boldsymbol{b}) = w_1 - (w_2 - w_1) = w_1 - s$ and we obtain

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = 2w_1 - \frac{w_1 - s}{2} = \frac{3w_1}{2} + \frac{s}{2}.$$

We observe that in this second case, we can have $\omega(\boldsymbol{a} * \boldsymbol{b}) = 0$. In this case, $w_2 = 2w_1$ and $\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = w_2$.

d. Given $\boldsymbol{a} \in C_{w_1}$ and $\boldsymbol{b} \in C_{w_2}$, we have either $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_1$ or $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_2$. If $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_1$,

$$w_1 = w_1 + w_2 - 2\omega(\boldsymbol{a} * \boldsymbol{b}),$$

i.e., $\omega(\boldsymbol{a} * \boldsymbol{b}) = \frac{w_2}{2}$. Then

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = w_1 + w_2 - \frac{w_2}{2} = w_1 + \frac{w_1 + s}{2} = \frac{3w_1}{2} + \frac{s}{2}.$$

In the second case,

$$w_2 = w_1 + w_2 - 2\omega(\boldsymbol{a} * \boldsymbol{b}).$$

That is, $\omega(\boldsymbol{a} * \boldsymbol{b}) = \frac{w_1}{2}$ and we obtain

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = w_1 + w_2 - \frac{w_1}{2} = w_1 + \frac{w_1}{2} + s = \frac{3w_1}{2} + s.$$

e. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are elements of C_{w_2} with $\boldsymbol{a} \neq \boldsymbol{b}$, then either $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_1$ or $\omega(\boldsymbol{a}+\boldsymbol{b}) = w_2$. In the first case,

$$w_1 = 2w_2 - 2\omega(\boldsymbol{a} * \boldsymbol{b}),$$

i.e., $\omega(\boldsymbol{a} * \boldsymbol{b}) = \frac{w_1}{2} + s$ and then

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = 2w_1 + 2s - \frac{w_1}{2} - s = \frac{3w_1}{2} + s.$$

For the last case,

$$w_2 = 2w_2 - 2\omega(\boldsymbol{a} * \boldsymbol{b}),$$

that is, $\omega(\boldsymbol{a} * \boldsymbol{b}) = \frac{w_2}{2}$ Then

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = 2w_2 - \frac{w_2}{2} = \frac{3w_2}{2}.$$

We have

Theorem 3.7. *Let $C_0 \subset \mathbb{F}_2^n$ be an $[n,\ k,\ d]$ linear code with two weights w_1 and w_2, where both are even, with $w_1 < w_2$ and $w_2 = w_1 + s$ for some even number s. Then for all $\boldsymbol{a} + \delta\boldsymbol{b} \in \mathbf{GU}(2,\ C_0) \setminus \{0\}$,*

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) \in \left\{ w_1,\ w_2,\ \frac{3w_1}{2},\ \frac{3w_2}{2},\ \frac{3w_1}{2} + \frac{s}{2},\ \frac{3w_1}{2} + s \right\},$$

where

$$\hat{A}_{w_1} = 3A_{w_1} \quad \textit{and} \quad \hat{A}_{w_2} \geq 3A_{w_2}.$$

Remark: In the particular case when $C_0 \subset \mathbb{F}_2^n$ is the binary first order Reed-Muller code $\boldsymbol{R}(1, m)$ with parameters $[2^m,\ m+1,\ 2^{m-1}]$, which is a **two-weight code** with $w_1 = 2^{m-1}$ and $w_2 = 2^m = 2w_1$, $\mathbf{GU}(2,\ C_0) \subset \mathbb{F}_4^{2^m}$ is a **three-weight code** and for all $\boldsymbol{a} + \delta\boldsymbol{b} \in \mathbf{GU}(2,\ C_0) \setminus \{0\}$,

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) \in \left\{ 2^{m-1},\ 2^{m-1} + 2^{m-2},\ 2^m \right\}.$$

We observe that our construction permits us to get examples of a three-weight code in an easier way comparing with the method used by K. Ding et al., in [22].

3.5.4 THREE-WEIGHT CODES FROM ANTIPODAL CODES

We get three-weight codes over $\mathbb{F}_{q^2}$ from the first-order generalized Reed-Muller code over $\mathbb{F}_q$, $\boldsymbol{R}_q(1, k)$, with parameters $[q^k,\ k+1,\ (q-1)q^{k-1}]$, see Theorem 2.7.

Foundation: Let $C \subset \mathbb{F}_q^n$ be an antipodal linear two-weight code and $C_0 \subset \mathbb{F}_q^n$ be the one-weight linear code generated by $(\boldsymbol{M}\ \mathbf{0})$, see Theorem 2.6. Then,

$$C = C_0 \cup (\boldsymbol{a}_1 + C_0) \cup \cdots \cup (\boldsymbol{a}_{q-1} + C_0)$$

with $\boldsymbol{a}_i \in C \setminus C_0$ and $1 \leq i \leq q-1$, i.e., $\omega(\boldsymbol{a}_i) = n$.

We suppose that C contains no codewords of full weight which are linearly independent, i.e., $\boldsymbol{a}_i = \lambda(1 \cdots 1) = \lambda \mathbf{1}$ for some $\lambda \in \mathbb{F}_q^*$, where $\omega(1 \cdots 1) = n$. We get

$$C = \cup_{\lambda \in \mathbb{F}_q} (\lambda \mathbf{1} + C_0),$$

and for all $\boldsymbol{a} \in C_0 \setminus \{0\}$ and for all $\lambda \in \mathbb{F}_q$,

$$\lambda \mathbf{1} + \boldsymbol{a} \in C_{w_1} = \{\boldsymbol{b} \in C \mid \omega(\boldsymbol{b}) = w_1\}$$

because otherwise, for some $\lambda \in \mathbb{F}_q^*$ and some $\boldsymbol{a} \in C_0 \setminus \{0\}$

$$\lambda \mathbf{1} + \boldsymbol{a} \in C_{w_2}.$$

That is, $\lambda \mathbf{1} + \boldsymbol{a} = \alpha \mathbf{1}$ for some $\alpha \in \mathbb{F}_q^*$ and $\boldsymbol{a} = (\alpha - \lambda)\mathbf{1} \in C_{w_2}$, which is not possible.

Now, for each $\lambda \in \mathbb{F}_q^*$, each $\boldsymbol{x} \in C_0$, from Equation (3.19),

$$n = \omega(\boldsymbol{x} + \delta \lambda \mathbf{1}) = \omega(\boldsymbol{x} + \lambda \mathbf{1}) + \omega(\boldsymbol{x} * \lambda \mathbf{1}),$$

i.e., $\omega(\boldsymbol{x} * \lambda \mathbf{1}) = n - d$. Then, $|\{i \mid x_i = -\lambda,\ \boldsymbol{x} \in C_0 \setminus \{0\}\}| = n - d$. In words, we have each $\boldsymbol{x} \in C_0 \setminus \{0\}$ contains $n - d$ times each $\lambda \in \mathbb{F}_q^*$. That is, $d = w_1 = (q-1)(n-d)$, i.e.,

$$n = \frac{qd}{q-1}$$

and we get that $(q-1) \mid d$. From Proposition 2.1, $w_1 = rq^{k-1}$, where r is the replication number of $S_k(q)$, i.e., $w_1 = q^{k-1}(q-1)w$ for some integer $w \geq 1$.

Since $n = \frac{qd}{q-1}$, we obtain

$$n = \frac{q}{q-1} q^{k-1}(q-1)w = q^k w.$$

Thus, C is a $[q^k w,\ k+1,\ (q-1)q^{k-1}w]$ code, see Theorem 2.7.

We apply our construction to the linear code C with parameters $[q^k,\ k+1,\ (q-1)q^{k-1}]$.

Since $\boldsymbol{x} \in C_0 \setminus \{0\}$ contains $n-d$ times each $\lambda \in \mathbb{F}_q^*$, the number of zero coordinates is $q^k - (q-1)q^{k-1} = q^{k-1} = n-d$, i.e., $\boldsymbol{x} \neq 0$ contains $n-d$ times each element of $\mathbb{F}_q$. We observe that given $\boldsymbol{a} \in S_k(q) \setminus \{0\}$ and $\mathbb{F}_q^* = \{1, \rho_1, \cdots, \rho_{q-2}\}$, if $a_i \neq 0$ then $\mathbb{F}_q^* = \{a_i, \rho_1 a_i, \cdots, \rho_{q-2} a_i\}$. Therefore, from Proposition 2.1, we can write

$$\boldsymbol{x} = (\boldsymbol{a}\ \rho_1 \boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0),$$

where $\boldsymbol{a} \in S_k(q) \setminus \{0\}$. That is, given $\boldsymbol{y} \in C$, there is $\lambda \in \mathbb{F}_q$ and $\boldsymbol{b} \in S_k(q)$, such that

$$\boldsymbol{y} = \lambda \mathbf{1} + (\boldsymbol{b}\ \rho_1 \boldsymbol{b}\ \cdots\ \rho_{q-2}\boldsymbol{b}\ 0).$$

Theorem 3.8. *Let $C \subset \mathbb{F}_q^n$ be a $[q^k,\ k+1,\ (q-1)q^{k-1}]$ antipodal linear two-weight code that contains no linearly independent codewords of full weight. Then, $\mathbf{GU}(2,\ C)$ is a three-weight code, where $w_1 = d = (q-1)q^{k-1}$, $w_2 = \frac{q+1}{q} w_1$, and $w_3 = n$.*

Proof. Given $\boldsymbol{x},\ \boldsymbol{y} \in C$, from Lemma 3.1,

$$\boldsymbol{x} + \delta \boldsymbol{y} = (1\ \delta) \begin{pmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{pmatrix}$$

and

$$\omega(\boldsymbol{x} + \delta \boldsymbol{y}) = n - |\{i \mid x_i = 0 = y_i\}|.$$

Since $C_0 = \{(\boldsymbol{a}\ \rho_1 \boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0) \mid \boldsymbol{a} \in S_k(q)\}$,

$$C = \{\lambda \mathbf{1} + \boldsymbol{x} \mid \lambda \in \mathbb{F}_q,\ \boldsymbol{x} \in C_0\} = \cup_\lambda (\lambda \mathbf{1} + C_0).$$

Let $\boldsymbol{x} = \alpha \mathbf{1} + (\boldsymbol{a}\ \rho_1 \boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ and $\boldsymbol{y} = \lambda \mathbf{1} + (\boldsymbol{b}\ \rho_1 \boldsymbol{b}\ \cdots\ \rho_{q-2}\boldsymbol{b}\ 0)$, where $\boldsymbol{a},\ \boldsymbol{b} \in S_k(q) \setminus \{0\}$ and $\{\boldsymbol{a},\ \boldsymbol{b}\}$ is a linearly independent set over $\mathbb{F}_q$. We want to calculate $|\{i :\ \mid x_i = 0 = y_i\}|$.

Since $\{\boldsymbol{a},\ \boldsymbol{b}\}$ is a linearly independent set, $(\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ and $(\boldsymbol{b}\ \rho_1\boldsymbol{b}\ \cdots\ \rho_{q-2}\boldsymbol{b}\ 0)$ are two linearly independent elements of C_0, we can take them as the first two rows of a generator matrix of C_0. We set the elements of the column j of the form $(-\alpha\ -\lambda\ c_{3j}\ \cdots\ c_{kj}) \in \mathbf{F}_q^k$, where $(c_{3j}\ \cdots\ c_{kj}) \in \mathbf{F}_q^{k-2}$. We obtain q^{k-2} such columns, i.e., $|\{i :\ | x_i = 0 = y_i\}| = q^{k-2}$. Then

$$\omega(\boldsymbol{x} + \delta\boldsymbol{y}) = n - q^{k-2} = \frac{q+1}{q}(q-1)q^{k-1}. \tag{3.20}$$

On the other hand, let $\boldsymbol{x} = \alpha\mathbf{1} + (\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ and $\boldsymbol{y} = \lambda\mathbf{1} + (\boldsymbol{b}\ \rho_1\boldsymbol{b}\ \cdots\ \rho_{q-2}\boldsymbol{b}\ 0)$, with $\boldsymbol{a} \neq 0,\ \boldsymbol{b} \neq 0$, and $\{\boldsymbol{a},\ \boldsymbol{b}\}$ linearly dependent. Let $\boldsymbol{b} = r\boldsymbol{a}$, for some $r \in \mathbb{F}_q^*$. If $\lambda = r\alpha$, i.e., $\boldsymbol{y} = r\boldsymbol{x}$ then

$$\omega(\boldsymbol{x} + \delta\boldsymbol{y}) = \omega(\boldsymbol{x}) = w_1 = (q-1)q^{k-1}.$$

If $\lambda \neq r\alpha$, when $x_i = 0$, $y_i \neq 0$, then

$$\omega(\boldsymbol{x} + \delta\boldsymbol{y}) = n.$$

We observe that $\omega(\alpha\mathbf{1}+\delta\boldsymbol{y}) = n = \omega(\boldsymbol{x}+\delta\lambda\mathbf{1})$ for all $\alpha,\ \lambda \in \mathbb{F}_q^*$ and for all $\boldsymbol{x},\ \boldsymbol{y} \in C$. Therefore, for all $\boldsymbol{x},\ \boldsymbol{y} \in C$

$$\omega(\boldsymbol{x} + \delta\boldsymbol{y}) \in \left\{0,\ w_1,\ \frac{q+1}{q}w_1,\ n\right\}.$$

□

Corollary 3.2. *The weight distribution of* $\mathbf{GU}(2,\ C)$ *is*

$$\hat{A}_{w_1} = (q+1)A_{w_1},$$

$$\hat{A}_{w_2} = A_{w_1}(A_{w_1} - q^2 + q),$$

and

$$\hat{A}_n = (q^2-1)(A_{w_1} + 1),$$

where $A_{w_1} = |\{\boldsymbol{x} \in C \mid \omega(\boldsymbol{x}) = w_1\}| = q^{k+1} - q$.

Proof. Since $C = \cup_\lambda(\lambda\mathbf{1} + C_0)$, we get

$$\mathbf{GU}(2,\ C) = \cup_{\lambda,\,\alpha}(\alpha\mathbf{1} + C_0 + \delta(\lambda\mathbf{1} + C_0)).$$

Let $\boldsymbol{x} = \alpha\mathbf{1} + (\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ and $\boldsymbol{y} = \lambda\mathbf{1} + (\boldsymbol{b}\ \rho_1\boldsymbol{b}\ \cdots\ \rho_{q-2}\boldsymbol{b}\ 0)$ be two elements in C such that $\boldsymbol{a},\ \boldsymbol{b} \in S_k(q) \setminus \{0\}$ and $\{\boldsymbol{a},\ \boldsymbol{b}\}$ is a linearly independent set over $\mathbb{F}_q$, i.e., $\boldsymbol{b} \notin \{\mathbf{0},\ \boldsymbol{a},\ \rho_1\boldsymbol{a},\ \cdots,\ \rho_{q-2}\boldsymbol{a}\}$. Then, the number of codewords $\boldsymbol{y}$ is $q^k - q$ for a fixed $\lambda \in \mathbb{F}_q$. The number of codewords $\boldsymbol{x}$ is $q^k - 1$ for a fixed α, and from Equation (3.20), $\omega(\boldsymbol{x} + \delta\boldsymbol{y}) = w_2$. Therefore, $(q^k - 1)(q^k - q)$ codewords for each λ. That is, we obtain

$$q(q^k - 1)(q^k - q) \tag{3.21}$$

codewords of weight w_2 for a fixed α. Thus, for any α and any λ,

$$\hat{A}_{w_2} = q[q(q^k - 1)(q^k - q)] = (q^{k+1} - q)(q^{k+1} - q^2) = A_{w_1}(A_{w_1} - q^2 + q). \tag{3.22}$$

On the other hand, observe that for any $\lambda \in \mathbb{F}_q^*$ and for all $\boldsymbol{x} \in C$, $\omega(\boldsymbol{x} + \delta\lambda\mathbf{1}) = n$, which gives $q^{k+1}(q-1)$ codewords of weight n. Now, for any $\lambda \in \mathbb{F}_q^*$ and for all $\boldsymbol{x} \in C \setminus \{\lambda\mathbf{1}\}$, again $\omega(\lambda\mathbf{1} + \delta\boldsymbol{x}) = n$, from which we get $(q-1)(q^{k+1} - (q-1))$ codewords of weight n.

If we take $\boldsymbol{x} = \alpha\mathbf{1} + (\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$, $\boldsymbol{y} = \lambda(\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ with $\boldsymbol{a} \neq 0$ and $\lambda \neq 0$, we know $\omega(\boldsymbol{x} + \delta\boldsymbol{y}) = n$. Then we obtain $(q-1)[(q-1)(q^k - 1)]$ codewords of weight n, we get the same number taking $\boldsymbol{x} = \lambda(\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ and $\boldsymbol{y} = \alpha\mathbf{1} + (\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$.

When $\boldsymbol{x} = \alpha\mathbf{1} + (\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ with $\boldsymbol{a} \neq 0$ and $\alpha \neq 0$, $\boldsymbol{y} = \alpha\mathbf{1} + \lambda(\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ with $\lambda \neq 0$ and $\lambda \neq 1$, we have $(q-1)[(q-2)(q^k - 1)]$ codewords of weight n.

Finally, taking $\boldsymbol{x} = \alpha\mathbf{1} + (\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ and $\boldsymbol{y} = \lambda\mathbf{1} + r(\boldsymbol{a}\ \rho_1\boldsymbol{a}\ \cdots\ \rho_{q-2}\boldsymbol{a}\ 0)$ with $\alpha,\ r,\ \lambda \in \mathbb{F}_q^*$, $\lambda \neq \alpha$, $\lambda \neq r\alpha$, i.e., $\boldsymbol{y} \neq r\boldsymbol{x}$ and $\boldsymbol{x} \neq 0$, we get $\omega(\boldsymbol{x} + \delta\boldsymbol{y}) =$

n, which gives us $(q^k-1)(q-2)$ codewords, for fixed λ and α. Then, we get $(q-1)[(q-2)[(q-2)(q^k-1)]]$.

Thus, letting $\Delta = q^k - 1$, we have

$$\hat{A}_n = (q-1)q^{k+1} + (q-1)\left(q^{k+1} - q + 1\right) + 2(q-1)^2\Delta + (q-1)(q-2)\Delta + (q-1)(q-2)^2\Delta.$$

That is,

$$\hat{A}_n = (q^2-1)(A_{w_1}+1). \tag{3.23}$$

Therefore,

$$\hat{A}_{w_1} = q^{2k+2} - 1 - \hat{A}_n - \hat{A}_{w_2} = (q+1)A_{w_1}. \tag{3.24}$$

Equations (3.22), (3.23) and (3.24) prove the corollary. □

Now, we take C_1 as the $r-$fold replication of the two-weight linear code in Theorem 3.8. If $\boldsymbol{y} \in C_1 \setminus \{0\}$, then there exists $\boldsymbol{x} \in C \setminus \{0\}$ such that

$$\boldsymbol{y} = (\boldsymbol{x}\ \alpha_1\boldsymbol{x}\ \cdots\ \alpha_{r-1}\boldsymbol{x}).$$

Since

$$\omega(\boldsymbol{x}) \in \left\{ w_1 = (q-1)q^{k-1}, w_2 = \frac{q+1}{q}w_1, w_3 = q^k \right\},$$

we get

$$\omega(\boldsymbol{y}) \in \left\{ w_1 = r(q-1)q^{k-1}, w_2 = \frac{q+1}{q}w_1, w_3 = rq^k \right\}.$$

That is,

Theorem 3.9. *Let $C \subset \mathbb{F}_q^n$ be a $[rq^k,\ k+1,\ r(q-1)q^{k-1}]$ antipodal linear two-weight code that contains no linearly independent codewords of full weight. Then, $\mathbf{GU}(2,\ C)$ is a three-weight code, where $w_1 = d = r(q-1)q^{k-1}$, $w_2 = \frac{q+1}{q}w_1$ and $w_3 = rq^k = n$.*

Example 3.12a. *Taking the binary case,* $C = \boldsymbol{R}(1,2)$*, and* $\mathbf{GU}(2,\ \boldsymbol{R}(1,2))$ *has* $4^3 = 64$ *codewords with the following weight distribution:*

$$\hat{A}_0 = 1,$$

$$\hat{A}_2 = 6 + 2 \times 6 = 3 \times 6 = 3(2^3 - 2) = 3A_2,$$

$$\hat{A}_{2^2} = 1 + 8 + 2 \times 6 = 3 \times 6 + 3 = 3(2^3 - 2) + 3 = 3A_2 + 3,$$

$$\hat{A}_3 = 6 \times 4 = (2^3 - 2)(2^3 - 4) = A_2(A_2 - 2).$$

We observe that $3 = 2^{2-1} + 2^{2-2}$.

b. *Taking* $k = 3$*, computing the* $4^4 = 256$ *codewords of* $\mathbf{GU}(2,\ \boldsymbol{R}(1,3))$ *we get the following weight distribution:*

$$\hat{A}_0 = 1,$$

$$\hat{A}_{2^2} = 42 = (2^4 - 2) \times 2 + (2^4 - 2) = 2A_2 + A_2 = 3A_{2^2},$$

$$\hat{A}_{2^3} = 45 = (2^4 - 2) \times 2 + 2^4 + 1 = (2^4 - 2) \times 2 + 2^4 - 2 + 3 = 3A_{2^2} + 3,$$

$$\hat{A}_6 = 168 = (2^4 - 2)(2^4 - 4) = (2^4 - 2)(2^4 - 2 - 2) = A_{2^2}(A_{2^2} - 2),$$

where $6 = 2^{3-1} + 2^{3-2}$.

c. *In general,* $\mathbf{GU}(2,\ \boldsymbol{R}(1,k))$ *has* 4^{k+1} *codewords and, from Corollary 3.2, its weight distribution is given by Table 3–10.*

w	$\hat{A}_w$
0	*1*
$\frac{n}{2}$	$3A_{\frac{n}{2}} = 3(2^{k+1} - 2)$
$\frac{3n}{4}$	$A_{\frac{n}{2}}(A_{\frac{n}{2}} - 2) = (2^{k+1} - 2)(2^{k+1} - 4)$
n	$3(A_{\frac{n}{2}} + 1) = 3(2^{k+1} - 1)$

Table 3–10: *Weight Distribution of* $\mathbf{GU}(2,\ \boldsymbol{R}(1,k))$

Theorem 3.10. *Using the notation of the Theorem 3.8,* $\mathbf{GU}(C, C_0)$ *is an additive three-weight code with*

$$w_1 = (q-1)q^{k-1}, \;\; w_2 = \frac{q+1}{q} w_1, \;\; w_3 = n = q^k$$

and its weight distribution is

$$\hat{A}_{w_1} = 2A_{w_1}, \; \hat{A}_{w_2} = \frac{1}{q} A_{w_1} \left(A_{w_1} - q^2 + q \right), \; \hat{A}_{w_3} = (q-1)(A_{w_1} + 1),$$

where $A_{w_1} = q^{k+1} - q$.

Proof. Since

$$C = \{\alpha \mathbf{1} + \boldsymbol{x} \mid \alpha \in \mathbb{F}_q, \; \boldsymbol{x} \in C_0\} = \cup_{\alpha \in \mathbb{F}_q} (\alpha \mathbf{1} + C_0)$$

we have

$$\mathbf{GU}(C, C_0) = \cup_{\alpha} (\alpha \mathbf{1} + C_0 + \delta C_0).$$

From Theorem 3.8, $\mathbf{GU}(C, C_0)$ is a three-weight additive code, with

$$w_1 = (q-1)q^{k-1}, \;\; w_2 = \frac{q+1}{q} w_1, \;\; w_3 = n = q^k.$$

From Equation (3.21),

$$\hat{A}_{w_2} = q(q-1)(q^k - q) = \frac{1}{q} A_{w_1} (A_{w_1} - q^2 + q).$$

On the other hand, let $\{\boldsymbol{a}, \boldsymbol{b}\} \subset S_k(q) \setminus \{0\}$ be a linearly dependent set over $\mathbb{F}_q$, i.e., $\boldsymbol{b} = \lambda \boldsymbol{a}$ for some $\lambda \in \mathbb{F}_q^*$. Then, for $\boldsymbol{x} = \alpha \mathbf{1} + (\boldsymbol{a} \; \rho_1 \boldsymbol{a} \; \cdots \; \rho_{q-2} \boldsymbol{a} \; 0) \in C$ and $\boldsymbol{y} = \lambda (\boldsymbol{a} \; \rho_1 \boldsymbol{a} \; \cdots \; \rho_{q-2} \boldsymbol{a} \; 0) \in C_0$, we have

$$\omega(\boldsymbol{x} + \delta \boldsymbol{y}) = n$$

and we get $(q^k - 1)(q-1)$ codewords for a fixed α, i.e., we obtain $(q-1)(q^k-1)(q-1)$ codewords with weight n. In addition, when $\boldsymbol{a} = \mathbf{0}$, $\omega(\alpha \mathbf{1} + \delta \boldsymbol{y}) = n$, this gives

$(q-1)q^k$ codewords. That is,

$$\hat{A}_{w_3} = (q-1)(q^k-1)(q-1) + (q-1)q^k = (q-1)(A_{w_1}+1).$$

Thus,

$$\hat{A}_{w_1} = q^{2k+1} - 1 - \hat{A}_{w_3} - \hat{A}_{w_2} = 2(q^{k+1} - q).$$

□

Example 3.13. *Taking* $q = 2$, $C_0 = \{(\boldsymbol{a}\ 0) \mid \boldsymbol{a} \in S_k(2)\}$. *Then* $\mathbf{GU}(\mathbf{R}(1,k), C_0)$ *is an additive three-weight code, with* $w_1 = 2^{k-1}$, $w_2 = \frac{3}{2}w_1$, *and* $w_3 = n = 2^k$. *Its weight distribution is* $\hat{A}_{w_1} = 2A_{w_1}$, $\hat{A}_{w_2} = \frac{1}{2}A_{w_1}(A_{w_1}-2)$, *and* $\hat{A}_{w_3} = A_{w_1}+1$, *where* $A_{w_1} = 2^{k+1} - 2$.

From Theorem 2.6, we can write $\mathbf{R}(1,k) = C_0 \cup (\mathbf{1} + C_0)$. *Then*

$$\mathbf{GU}(\mathbf{R}(1,k), C_0) = \mathbf{GU}(C_0, C_0) \cup (\mathbf{1} + C_0 + \delta C_0). \tag{3.25}$$

We know, for any nonzero $\boldsymbol{x} \in \mathbf{GU}(C_0, C_0)$, $\omega(\boldsymbol{x}) \in \left\{2^{k-1},\ 2^{k-1} + 2^{k-2}\right\}$, *see Theorem 3.6.*

On the other hand, let $\boldsymbol{x} = \mathbf{1} + (\boldsymbol{a}\ 0) \in \mathbf{1} + C_0$ *and* $\boldsymbol{y} = (\boldsymbol{b}\ 0) \in C_0$. *If* $\{\boldsymbol{a}, \boldsymbol{b}\}$ *is a linearly independent set, from Equation (3.20),*

$$\omega(\boldsymbol{x} + \delta\boldsymbol{y}) = n - 2^{k-2} = 2^{k-1} + 2^{k-2}.$$

But, if $\{\boldsymbol{a}, \boldsymbol{b}\}$ *is a linearly dependent set, we have*

- *If* $\boldsymbol{a} = \boldsymbol{b}$, *then* $\omega(\mathbf{1} + (\boldsymbol{a}\ 0) + \delta(\boldsymbol{a}\ 0)) = n = 2^k$.
- *If* $\boldsymbol{a} = 0$ *and* $\boldsymbol{b} \neq 0$, *then* $\omega(\mathbf{1} + \delta(\boldsymbol{b}\ 0)) = n = 2^k$.
- *If* $\boldsymbol{a} \neq 0$ *and* $\boldsymbol{b} = 0$, *then* $\omega(\mathbf{1} + (\boldsymbol{a}\ 0) + \delta\mathbf{0}) = 2^{k-1}$.

Therefore, for any nonzero $\boldsymbol{x}$, $\boldsymbol{y}$ *in* $\mathbf{GU}(\mathbf{R}(1,k), C_0)$,

$$\omega(\boldsymbol{x} + \delta\boldsymbol{y}) \in \left\{2^{k-1}, 2^{k-1} + 2^{k-2}, 2^k\right\}$$

Considering Equation (3.25) and when $\{\boldsymbol{a}, \boldsymbol{b}\} \subset S_k(2)$ *is a linearly independent or not, we get that* $\hat{A}_{w_1} = 2(2^{k+1}-2)$, $\hat{A}_{w_2} = \frac{1}{2}(2^{k+1}-2)(2^{k+1}-4)$, *and* $\hat{A}_{w_3} = 2^{k+1}-1$.

An observation: We note that taking $k = 3$ we obtain the optimal parameters of the quaternary additive code of length 8, i.e., the additive code $\mathbf{GU}(\mathbf{R}(1,3), C_0)$ has parameters $[2^3, 2^7, 2^2]$, see *Table* 1 in [8] .

CHAPTER 4
QUANTUM CODES FROM THE GO-UP CONSTRUCTION

In this chapter we present a relation between our construction and quantum stabilizer codes. In the first section, given a code over $\mathbb{F}_4$ that is Euclidean self-orthogonal, we get a quantum stabilizer code. See Proposition 4.1, Theorem 4.1 and their applications. In these theorems, our choice of Euclidean duality is important. In **Section 4.2** we show that for an amalgamated code C in $\mathbb{F}_{q^2}^n$, Euclidean duality is the same as the trace-alternating duality considered in [31]. We use this result to construct nonbinary additive stabilizer codes using Euclidean duality and Frobenius invariant codes, see Theorem 4.2. We give another way to construct q-ary **QECC**. Calderbankd and Shor [14] obtain **QECC** from binary quantum codes by showing that under their conditions, Hermitian and Trace Hermitian inner products are equivalent. Using our **Go-Up** construction, we are able to obtain the same equivalence of these inner products for q-ary codes when q is a prime such that $q \equiv 3 \pmod 4$, using Theorem 3.5. Therefore, with our Go-Up construction, we can obtain q-ary **QECC**.

4.1 A REFORMULATION OF BINARY STABILIZER CODES AND NEW CONSTRUCTIONS OF QUANTUM ERROR-CORRECTING CODES FROM THE GO-UP CONSTRUCTION

We recall a postulate of quantum physics which states that quantum evolution is unitary, that is, if we have some arbitrary quantum operator U that takes as input state $|\boldsymbol{u}>$ and output a different state $U|\boldsymbol{u}>$, then we may describe U as a unitary operator, see Definition 2.10.

We define

$$\xi_n = \{i^r w_1 \otimes \cdots \otimes w_n | w_j \in \{I, \sigma_x, \sigma_z, \sigma_y\}, r \in \mathbb{Z}_4\},$$

a subgroup of the unitary group $U(2^n)$, ξ_n is the tensor product of all possible qubit errors, that is, ξ_n describes a set of discrete errors in n qubits.

Since $w_j \in \{I, \sigma_x, \sigma_z, \sigma_y\}$ and $r \in \mathbb{Z}_4$, $|\xi_n| = 4 \times 4^n = 2^{2n+2}$. The center of ξ_n is $Z(\xi_n) = \{i^r I | r \in \mathbb{Z}_4\}$, hence $|Z(\xi_n)| = 4$. Taking $\overline{\xi_n} = \xi_n / Z(\xi_n)$, we have $|\overline{\xi_n}| = 2^{2n+2}/4 = 2^{2n}$. Indeed we have

$$\overline{\xi_n} \cong \mathbb{F}_2^{2n}$$

via identification

$$I \leftrightarrow (00) \;\; \sigma_x \leftrightarrow (10)$$

$$\sigma_z \leftrightarrow (01) \;\; \sigma_y \leftrightarrow (11),$$

where multiplication of matrices in $\overline{\xi_n}$ corresponds to the addition of vectors in $\mathbb{F}_2^{2n}$. That is, $\overline{\xi_n}$ is an abelian group of order 2^{2n} and therefore a binary vector space.

Observe that to each element of $\overline{\xi_n}$ corresponds a vector $\boldsymbol{a} + \delta\boldsymbol{b} \in \mathbb{F}_4^n$ and two such vectors $\boldsymbol{x} = \boldsymbol{a} + \delta\boldsymbol{b}$ and $\boldsymbol{y} = \boldsymbol{c} + \delta\boldsymbol{d}$ come from commutative operators if and only if $\boldsymbol{x} \cdot_{TH} \boldsymbol{y} = \boldsymbol{a} \cdot \boldsymbol{d} + \boldsymbol{b} \cdot \boldsymbol{c} = 0$, see Proposition 2.5 and Equation (3.14).

Example 4.1a. *Taking $n = 1$, let $\xi_1 = \{i^r w | w \in \{I, \sigma_x, \sigma_z, \sigma_y\}, r \in \mathbb{Z}_4\}$.*

$$\sigma_x \sigma_z = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = -i \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = -i\sigma_y$$

that is,

$$\sigma_x \sigma_z \equiv \sigma_y$$

in $\overline{\xi_1}$, and we can see using the identification that

$$(10) + (01) = (11).$$

b. *Taking* $n = 2$, *let* $\xi_2 = \{i^r w_1 \otimes w_2 | w_j \in \{I, \sigma_x, \sigma_z, \sigma_y\}, r \in \mathbb{Z}_4\}$

$$(\sigma_x \otimes \sigma_y)(\sigma_x \otimes \sigma_z) = \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 0 & i & 0 \end{pmatrix}.$$

That is,

$$(\sigma_x \otimes \sigma_y)(\sigma_x \otimes \sigma_z) = i \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = i(I \otimes \sigma_x).$$

Then,

$$(\sigma_x \otimes \sigma_y)(\sigma_x \otimes \sigma_z) \equiv I \otimes \sigma_x$$

in $\overline{\xi_2}$ *and we observe that by identification we get*

$$(1\ 0\ 1\ 1) + (1\ 0\ 0\ 1) = (0\ 0\ 1\ 0).$$

Definition 4.1. *Given* $\mathbb{S} = \{E = i^r U_{\boldsymbol{a}} V_{\boldsymbol{b}} \in \xi_n \mid \boldsymbol{a} + \delta \boldsymbol{b} \in \mathbb{F}_4^n,\ r \in \mathbb{Z}_4\}$, *the centralizer of* $\mathbb{S}$ *in* ξ_n *with respect to the Trace Hermitian inner product is defined by*

$$\mathbb{S}^{\perp_{TH}} = \{E_1 = i^{r_1} U_{\boldsymbol{c}} V_{\boldsymbol{d}} \in \xi_n \mid \boldsymbol{c} + \delta \boldsymbol{d} \in \mathbb{F}_4^n,\ r_1 \in \mathbb{Z}_4 \text{ and } \boldsymbol{a} \cdot \boldsymbol{d} = \boldsymbol{b} \cdot \boldsymbol{c}\}$$

the subset of all unitary operators in ξ_n *which commute with all the elements of* $\mathbb{S}$.

Now, we are going to show a fruitful relation between binary quantum stabilizer codes and our **Go-Up** construction.

Proposition 4.1. *Let* $C_0 \subset \mathbb{F}_2^n$ *be an* $[n,\ k,\ d]$ *Euclidean self-orthogonal linear code. Then,* $C = \mathbf{GU}(2,\ C_0) \subset \mathbb{F}_4^n$ *is Euclidean self-orthogonal and we get an* $[[n,\ n-2k, d]]_2$ *quantum stabilizer code, where* d *is the Hamming minimum weight of* $C_0^{\perp} \setminus C_0$.

Proof. For any two elements from

$$\mathbb{S} = \{E = i^r U_{\mathbf{a}} V_{\mathbf{b}} \mid \boldsymbol{a} + \delta \boldsymbol{b} \in \mathbf{GU}(2,\ C_0),\ r \in \mathbb{Z}_4\}$$

$E = i^r U_{\mathbf{a}} V_{\mathbf{b}}$ and $E_1 = i^{r_1} U_{\mathbf{c}} V_{\mathbf{d}}$, from Equations (2.6), (2.7) and (2.8), we have

$$EE_1|\mathbf{v}>= Ei^{r_1}(-1)^{\mathbf{d}\cdot\mathbf{v}}|\mathbf{v}+\mathbf{c}>= i^{r+r_1}(-1)^{\mathbf{d}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c}}|\mathbf{v}+\mathbf{c}+\mathbf{a}>$$

and

$$E_1E|\mathbf{v}>= i^{r_1+r}(-1)^{\mathbf{b}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{a}}|\mathbf{v}+\mathbf{a}+\mathbf{c}> .$$

Then,

$$EE_1 = E_1E$$

when

$$\mathbf{d}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c} = \mathbf{b}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{v}+\mathbf{a}\cdot\mathbf{d}$$

for any $|\mathbf{v}>$, that is, when

$$\mathbf{a}\cdot\mathbf{d} = \mathbf{b}\cdot\mathbf{c}.$$

Since $C_0 \subset C_0^{\perp}$,

$$\mathbf{a}\cdot\mathbf{d} = 0 = \mathbf{b}\cdot\mathbf{c},$$

and we have proven that the operators E and E_1 commute.

We observe that $EE_1|\mathbf{v}>= i^{r+r_1}(-1)^{(\mathbf{b}+\mathbf{d})\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c}}|\mathbf{v}+\mathbf{a}+\mathbf{c}>= i^s U_{\mathbf{a}+\mathbf{c}} V_{\mathbf{b}+\mathbf{d}}|\mathbf{v}>$, because $\mathbf{b}\cdot\mathbf{c} = 0$, where $s = r + r_1 \in \mathbb{Z}_4$, then $EE_1 \in \mathbb{S}$. Since $E^{-1} = i^{-r} U_{\mathbf{a}} V_{\mathbf{b}}$, $E^{-1} \in \mathbb{S}$ and $E = U_{\mathbf{0}} V_{\mathbf{0}} \in \mathbb{S}$ is the identity element, we get that $\mathbb{S}$ is an abelian subgroup of ξ_n such that $\overline{\mathbb{S}} = \mathbf{GU}(2,\ C_0) \subset \mathbb{F}_4^n$ has $2^{2k} = 2^{n-(n-2k)}$ codewords and it is an Euclidean self-orthogonal linear code by *Proposition* 3.5.

Now, from *Proposition* 3.7, we can take

$$\mathbb{S}^{\perp_{TH}} = \left\{ g = i^r U_{\mathbf{a}} V_{\mathbf{b}} \in \xi_n | \boldsymbol{a} + \delta \boldsymbol{b} \in C_0^{\perp} + \delta C_0^{\perp} \right\},$$

where for any $g = i^r U_{\mathbf{a}} V_{\mathbf{b}}$ and $g_1 = i^{r_1} U_{\mathbf{c}} V_{\mathbf{d}}$ in $\mathbb{S}^{\perp_{TH}}$,

$$gg_1|\mathbf{v}>= i^{r+r_1}(-1)^{\mathbf{d}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c}}|\mathbf{v}+\mathbf{c}+\mathbf{a}>= (-1)^{\boldsymbol{b}\cdot\boldsymbol{c}} i^{r+r_1} U_{\boldsymbol{a}+\boldsymbol{c}} V_{\boldsymbol{b}+\boldsymbol{d}}|\boldsymbol{v}> .$$

If $\boldsymbol{b}\cdot\boldsymbol{c}$ is even, $(-1)^{\boldsymbol{b}\cdot\boldsymbol{c}} = 1$ and $gg_1 \in \mathbb{S}^{\perp_{TH}}$. If $\boldsymbol{b}\cdot\boldsymbol{c}$ is odd number, $(-1)^{\boldsymbol{b}\cdot\boldsymbol{c}} = -1 = i^2$, and again $gg_1 \in \mathbb{S}^{\perp_{TH}}$. That is, $\mathbb{S}^{\perp_{TH}}$ is the subgroup of ξ_n such that for all $g \in \mathbb{S}^{\perp_{TH}}$ and any $E \in \mathbb{S}$, $gE = Eg$. Since $C \subset C^{\perp}$, $\mathbb{S} \subset \mathbb{S}^{\perp_{TH}}$. We take

$$\overline{\mathbb{S}}^{\perp_{TH}} = C_0^{\perp} + \delta C_0^{\perp} = C^{\perp}$$

having $2^{n-k}2^{n-k} = 2^{2n-2k}$ vectors over $\mathbb{F}_4$.

Now, we take $C_0^{\perp} + \delta C_0^{\perp} \setminus C_0 + \delta C_0 = \left\{\boldsymbol{a} + \delta\boldsymbol{b} \in C_0^{\perp} + \delta C_0^{\perp} | \mathbf{a},\ \mathbf{b} \in C_0^{\perp} \setminus C_0\right\}$. We know

$$\omega(\boldsymbol{a} + \delta\boldsymbol{b}) = \omega(\mathbf{a}) + \omega(\mathbf{b}) - \omega(\mathbf{a} * \mathbf{b}),$$

where $\mathbf{a} * \mathbf{b} = (c_1\ \cdots\ c_n)$ is such that $c_j = 1$ if and only if $a_j = b_j = 1$.

If $d = dist(C_0^{\perp} \setminus C_0)$, for all $\boldsymbol{a} + \delta\boldsymbol{b} \in C^{\perp} \setminus C$, we have $\omega(\boldsymbol{a} + \delta\boldsymbol{b}) \geq d$, i.e., there is no word $\mathbf{v} \in C^{\perp} \setminus C$ such that $\omega(\boldsymbol{v}) < d$. Thus, from Theorem 2.16, we get an $[[n,\ n-2k,\ d]]_2$ **linear quantum code**.

□

Observe that in *Proposition* 4.1 we are using Theorem 2.16, where $C \subset \mathbb{F}_4^n$ is self-orthogonal with respect to the Trace Hermitian inner product, but we can take $C = \mathbf{GU}(2,\ C_0) \subset \mathbb{F}_4^n$ **being Euclidean self-orthogonal** because of *Proposition* 3.7.

Theorem 4.1. *Let $C \subset \mathbb{F}_4^n$ be an $[n,\ k,\ d]_4$ Euclidean self-orthogonal linear code such that $C^2 \subset C$. Then, C yields an $[[n,\ n-2k,\ \geq d^{\perp}]]_2$ quantum stabilizer code that is pure to d.*

Proof. Let C_0 be the sub-field sub-code of C. Letting $q = 2$, from *Lemma* 3.2,

$$C = \mathbf{GU}(2,\ C_0),$$

and from *Proposition* 3.5,

$$C^{\perp} = \mathbf{GU}(2,\ C_0^{\perp}).$$

Since $C \subset C^{\perp}$, C_0 is Euclidean self-orthogonal and $dist(C^{\perp} \setminus C) \geq dist(C^{\perp}) = d^{\perp}$, now we apply *Proposition* 4.1. □

a. We have a quantum stabilizer code mapping $n-2k$ qubits to n qubits. From this, $n-2k \geq 0$, that is,

$$\frac{k}{n} \leq \frac{1}{2}$$

i.e., the rate of the linear code C_0, R, is $R \leq \frac{1}{2}$.

Example 4.2. *If $C_0 = \boldsymbol{R}(1,m)$ is the first-order Reed-Muller code of parameters $[2^m,\ m+1,\ 2^{m-1}]$, we have C_0 is self-orthogonal because $C_0^{\perp} = \boldsymbol{R}(m-2,m)$, $C_0^{\perp}$ has parameters $[2^m,\ 2^m-m-1,\ 4]$, and $\frac{m+1}{2^m} \leq \frac{1}{2}$ for all $m \geq 3$. That is,*

$$R_{\boldsymbol{R}(1,m)} \leq \frac{1}{2}\ \ \forall\ m \geq 3.$$

Thus, from the first-order Reed-Muller code, $C_0 = \boldsymbol{R}(1,m)$, we obtain the binary quantum code with parameters $[[2^m,\ 2^m-2m-2,\ 4]]_2$. We obtain $d=4$ because $d = dist(C_0^{\perp} \setminus C_0) = dist(C_0^{\perp}) = 4$, for all $m \geq 3$. Since the parameters satisfy the quantum Singleton bound

$$2(d-1) \leq n-k.$$

In this case, $n-k = 2m+2$, that is,

$$4 \leq m+2.$$

For $m=3$, we obtain $[[8,\ 0,\ 4]]_2$ from the Euclidean self-dual binary code $C_0 = \boldsymbol{R}(1,3)$. Another form to see that $d=4$, when $m=3$, is because $d \leq 5$ and it is known that, except for trivial codes, codes with $d \leq 2$, there are only two binary quantum MDS codes, $[[5,1,3]]_2$ and $[[6,0,4]]_2$. See Theorem 24 from [14].

We observe that we are using the Euclidean inner product in contrast with the seminal work [14] where they use the Trace Hermitian inner product.

Example 4.3. *Consider $H_k(2)$, the binary Hamming code with parameters $[2^k - 1,\ 2^k-1-k,\ 3]$, and C_0 as its dual with parameters $[2^k-1,\ k,\ 2^{k-1}]$, i.e., $C_0 = S_k(2)$ the binary simplex code of dimension k. Since the columns of a parity check matrix for a binary Hamming code consist of all possible nonzero binary words of length k, that is, all the elements of $\mathbb{F}_2^k \setminus \{0\}$, because $S_k(2)$ is a projective code; we can take α a primitive element of $\mathbb{F}_{2^k}$ and such parity check matrix is given by*

$$H = [\alpha^{2^k-2}, \cdots, \alpha,\ 1]$$

changing the order if necessary. Then

$$HH^t = \sum_{i=0}^{2^k-2} \left(\alpha^i\right)^2 = \left(\sum_{i=0}^{2^k-2} \alpha^i\right)^2,$$

because the characteristic is 2. But $\sum_{i=0}^{2^k-2} \alpha^i = 0$ because $0 = \alpha^{p^r} - \alpha = \alpha(\alpha - 1)(\alpha^{p^r-2} + \cdots + \alpha + 1)$ with $\alpha \neq 0$ and $\alpha - 1 \neq 0$, in our case $p = 2$. Thus,

$$HH^t = 0 \tag{4.1}$$

and since H is a generator matrix of $S_k(2)$, the binary simplex code of dimension k is Euclidean self-orthogonal with rate $\frac{k}{2^k-1} \leq \frac{1}{2}$ for all $k \geq 3$. That is,

$$R_{C_0} \leq \frac{1}{2} \ \ \forall\ k \geq 3$$

and since $dist(H_k(2) \setminus S_k(2)) = 3$, we obtain a binary stabilizer code of parameters $[[2^k - 1,\ 2^k - 1 - 2k,\ 3]]$. These parameters satisfy the quantum Singleton bound, i.e.,

$$3 - 1 \leq \frac{1}{2}(2^k - 1 - (2^k - 1 - 2k)) = k.$$

For the particular case $k = 3$, consider $C_0^{\perp}$ as the binary Hamming code of parameters $[7,\ 4,\ 3]$. We know $C_0 \subset C_0^{\perp}$. If we take a parity check matrix of $C_0^{\perp}$ given by

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$

then $C_0 = \{c = (a_1\ a_2\ a_3)H \mid a_i \in \mathbb{F}_2\} = S_3(2)$.

Thus, we obtain the linear quantum code with parameters $[[7,\ 1,\ 3]]$ which maps $7 - 2(3) = 1$ qubits into 7 qubits, see [15].

An observation: Consider the Goppa code $\mathbf{\Gamma}(L, g(x))$ with parameters $[n,\ k \geq n - mt,\ d]_2$. Since $k \geq n - mt$, $\frac{k}{n} \geq 1 - \frac{mt}{n}$. If $\frac{k}{n} \leq \frac{1}{2}$, $1 - \frac{mt}{n} \leq \frac{1}{2}$, that is, $2mt \geq n$. Thus, if $R_{\mathbf{\Gamma}(L,g(x))} \leq \frac{1}{2}$ then $2mt \geq n$, or equivalent if $2mt < n$, then $R_{\mathbf{\Gamma}(L,g(x))} > \frac{1}{2}$.

Corollary 4.1. *We take $h(x) = x^t$, $L = \{1, \alpha, \cdots, \alpha^{n-1}\} \subset \mathbb{F}_{2^m}$, where $\alpha \in \mathbb{F}_{2^m}$ and $ord(\alpha) = n$. If $t < \frac{n}{2}$, then we will obtain a linear quantum code with parameters $[[n,\ n - 2k_{\mathbf{\Gamma}(L,x^t)^{\perp}},\ d]]_2$, where $d = dist(\mathbf{\Gamma}(L, x^t) \setminus \mathbf{\Gamma}(L, x^t)^{\perp})$.*

Proof. A parity check matrix of $\mathbf{\Gamma}(L,\ h(x))$ is given by

$$H = \begin{pmatrix} 1 & \alpha^{t-1} & \alpha^{2(t-1)} & \cdots & \alpha^{(n-1)(t-1)} \\ 1 & \alpha^{t-2} & \alpha^{2(t-2)} & \cdots & \alpha^{(n-1)(t-2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} h^{-1}(1) & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & h^{-1}(\alpha^{n-1}) \end{pmatrix}.$$

Then,

$$H = \begin{pmatrix} 1 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(n-1)} \\ 1 & \alpha^{-2} & \alpha^{-4} & \cdots & \alpha^{-2(n-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{-t} & \alpha^{-2t} & \cdots & \alpha^{-(n-1)t} \end{pmatrix}$$

and

$$(HH^t)_{ij} = h^{-1}(1)h^{-1}(1)+h^{-1}(\alpha)\alpha^i h^{-1}(\alpha)\alpha^j+\cdots+h^{-1}(\alpha^{n-1})\alpha^{(n-1)i}h^{-1}(\alpha^{n-1})\alpha^{(n-1)j}$$

i.e., $(HH^t)_{ij} = 1 + \alpha^{-(i+j)} + (\alpha^{-(i+j)})^2 + \cdots + (\alpha^{-(i+j)})^{n-1}$, where $2 \leq i + j \leq 2t$. Since $t < \frac{n}{2}$, $\alpha^{-(i+j)} \neq 1$ and

$$(HH^t)_{ij} = (\alpha^{-(i+j)} - 1)^{-1}((\alpha^{-(i+j)})^n - 1) = 0.$$

Thus, $HH^t \equiv 0$ and

$$\mathbf{\Gamma}(L, x^t)^{\perp} \subset \mathbf{\Gamma}(L, x^t).$$

Taking

$$d = dist(\mathbf{\Gamma}(L, x^t) \setminus \mathbf{\Gamma}(L, x^t)^{\perp})$$

and $C_0 = \mathbf{\Gamma}(L, x^t)^{\perp}$, we get the stabilizer group $C = \mathbf{GU}(2,\ C_0)$ which permits us to construct a linear quantum code with parameters $[[n,\ n - 2k_{\mathbf{\Gamma}(L,x^t)^{\perp}},\ d]]_2$.

□

Now we consider different codes.

Proposition 4.2. *Let $C_0 \subset \mathbb{F}_2^n$ be an $[n,\ k,\ d]$ Euclidean self-orthogonal linear code. The additive code $C = C_0 + \delta C_0^{\perp} \subset \mathbb{F}_4^n$ is self-dual with respect to the Trace Hermitian inner product and we get a pure $[[n,\ 0,\ d]]_2$ quantum stabilizer code.*

Proof. We define

$$\mathbb{S} = \left\{E = i^r U_{\boldsymbol{a}} V_{\boldsymbol{b}} \in \xi_n | \mathbf{a} + \delta\mathbf{b} \in C_0 + \delta C_0^{\perp},\ r \in \mathbb{Z}_4\right\}$$

which has $|C_0 + \delta C_0^{\perp}| = 2^k 2^{n-k} = 2^n$ elements. For $E = i^r U_{\mathbf{a}} V_{\mathbf{b}}$ and $E_1 = i^{r_1} U_{\mathbf{c}} V_{\mathbf{d}}$ in $\mathbb{S}$, we have

$$EE_1|\boldsymbol{v}>= i^{r+r_1}(-1)^{(\mathbf{b}+\mathbf{d}).\mathbf{v}+\mathbf{b}.\mathbf{c}}|\mathbf{v}+\mathbf{a}+\mathbf{c}>= i^s U_{\mathbf{a}+\mathbf{c}} V_{\mathbf{b}+\mathbf{d}}|\mathbf{v}>$$

and

$$E_1E|\boldsymbol{v}>= i^{r+r_1}(-1)^{(\mathbf{b}+\mathbf{d}).\mathbf{v}+\mathbf{a}.\mathbf{d}}|\mathbf{v}+\mathbf{a}+\mathbf{c}>= i^s U_{\mathbf{a}+\mathbf{c}} V_{\mathbf{b}+\mathbf{d}}|\mathbf{v}>= EE_1|\boldsymbol{v}> .$$

Since $EE_1 \in \mathbb{S}$ and $E^{-1} = i^{-r} U_{\mathbf{a}} V_{\mathbf{b}} \in \mathbb{S}$, we get that $\mathbb{S}$ is an abelian subgroup of ξ_n and $\overline{\mathbb{S}} = C_0 + \delta C_0^{\perp} = C$.

Since $C_0 \subset C_0^{\perp}$, $C^{\perp_{TH}} = C_0 + \delta C_0^{\perp} = C$ and we apply the Theorem 2.16 wtih $k = 0$ to get a single quantum state pure code with parameters $[[n,\ 0,\ d]]_2$. □

Example 4.4. *From Example 3.9 we have the additive code* $C = S_k(2) + \delta H_k(2)$. *Then we get a single quantum state code with parameters* $[[2^k - 1,\ 0,\ 2^{k-1}]]_2$.

An observation: This is the same result as the one obtained in *Theorem* 9 from the seminal work [14], by taking $C_1 = C_0 = C_2$. However, there, they use a normal basis $\{\delta,\ \delta^2\}$ of $\mathbb{F}_4$ over $\mathbb{F}_2$, while here we use a polynomial basis $\{1,\ \delta\}$, where $\delta^2 = \delta + 1$.

Proposition 4.3. *Given* C_0 *an* $[n,\ k_0,\ d_0]_2$ *code and* C_1 *an* $[n,\ k_1,\ d_1]_2$ *code such that* $C_0 \subset C_1^{\perp}$, *we get an* $[[n,\ n-(k_0+k_1),\ d]]_2$ *quantum stabilizer code, with* $d = dist\left\{\mathbf{GU}(C_1^{\perp}, C_0^{\perp}) \setminus \mathbf{GU}(C_0, C_1)\right\}$.

Proof. We define

$$\mathbb{S} = \left\{E = i^r U_{\boldsymbol{a}} V_{\boldsymbol{b}} \in \xi_n | \mathbf{a} + \delta\mathbf{b} \in C_0 + \delta C_1,\ r \in \mathbb{Z}_4\right\}.$$

That is, $\overline{\mathbb{S}} = C_0 + \delta C_1$. For any two elements of $\mathbb{S}$, $E = i^r U_{\mathbf{a}} V_{\mathbf{b}}$ and $E_1 = i^{r_1} U_{\mathbf{c}} V_{\mathbf{d}}$ we have

$$EE_1|\mathbf{v}>= Ei^{r_1}(-1)^{\mathbf{d}.\mathbf{v}}|\mathbf{v}+\mathbf{c}>= i^{r+r_1}(-1)^{\mathbf{d}.\mathbf{v}+\mathbf{b}.\mathbf{v}+\mathbf{b}.\mathbf{c}}|\mathbf{v}+\mathbf{c}+\mathbf{a}>$$

and

$$E_1 E|\mathbf{v}>= i^{r_1+r}(-1)^{\mathbf{b}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{a}}|\mathbf{v}+\mathbf{a}+\mathbf{c}> .$$

Then,

$$EE_1 = E_1 E$$

when

$$\mathbf{d}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c}=\mathbf{b}\cdot\mathbf{v}+\mathbf{d}\cdot\mathbf{v}+\mathbf{a}\cdot\mathbf{d}$$

for any $|\mathbf{v}>$, that is, when

$$\mathbf{a}\cdot\mathbf{d}=\mathbf{b}\cdot\mathbf{c}.$$

Since $C_0 \subset C_1^{\perp}$,

$$\mathbf{a}\cdot\mathbf{d}=0=\mathbf{b}\cdot\mathbf{c}$$

and we have proven that the operators E and E_1 commute and $\overline{\mathbb{S}} \subset \overline{\mathbb{S}}^{\perp_{TH}}$.

We observe that $EE_1|\mathbf{v}>= i^{r+r_1}(-1)^{(\mathbf{b}+\mathbf{d})\cdot\mathbf{v}+\mathbf{b}\cdot\mathbf{c}}|\mathbf{v}+\mathbf{a}+\mathbf{c}>= i^s U_{\mathbf{a}+\mathbf{c}}V_{\mathbf{b}+\mathbf{d}}|\mathbf{v}>$, because $\mathbf{b}\cdot\mathbf{c}=0$, where $s=r+r_1 \in \mathbb{Z}_4$, then $EE_1 \in \mathbb{S}$. Since $E^{-1}=i^{-r}U_{\mathbf{a}}V_{\mathbf{b}}$, $E^{-1}\in\mathbb{S}$ and $E=U_{\mathbf{0}}V_{\mathbf{0}}\in\mathbb{S}$ is the identity element, we get that $\mathbb{S}$ is an abelian subgroup of ξ_n.

In this case we take

$$\mathbb{S}^{\perp_{TH}}=\left\{g=i^r U_{\boldsymbol{a}}V_{\boldsymbol{b}}\in\xi_n | \mathbf{a}+\delta\mathbf{b}\in C_1^{\perp}+\delta C_0^{\perp}\right\}.$$

Since $k_0 = dim(C_0)$ and $k_1 = dim(C_1)$, we get that $|\overline{\mathbb{S}}| = 2^{k_0+k_1} = 2^{n-(n-k_0-k_1)}$.

But

$$C_1^{\perp}+\delta C_0^{\perp}\setminus C_0+\delta C_1=\left\{\mathbf{a}+\delta\mathbf{b}\in C_1^{\perp}+\delta C_0^{\perp} \mid a\in C_1^{\perp}\setminus C_0,\ \boldsymbol{b}\in C_0^{\perp}\setminus C_1\right\}.$$

Then, $d = min\left\{dist(C_1^{\perp}\setminus C_0),\ dist(C_0^{\perp}\setminus C_1)\right\} = dist(C_1^{\perp}+\delta C_0^{\perp}\setminus C_0+\delta C_1)$. Thus, by Theorem 2.16 , we get an $[[n,\ n-(k_0+k_1),\ d]]_2$ additive quantum code.

□

Example 4.5. *From Example 3.5 we have the additive code* $C = \mathbf{GU}(C_0, C_1)$*, where* C_0 *is the repetition code with parameters* $[2^m,\ 1,\ 2^m]$ *and* C_1 *the first-order Reed-Muller code with parameters* $[2^m,\ m+1,\ 2^{m-1}]$*. Since* $C_0 \subset C_1 \subset C_1^{\perp}$*, we get a* $[[2^m,\ 2^m - m - 2,\ d]]_2$ *additive quantum code.*

4.2 NON-BINARY STABILIZER CODES

Given $\boldsymbol{a},\ \boldsymbol{b},\ \boldsymbol{c}$ and $\boldsymbol{d}$ in $\mathbb{F}_q^n$, let $\boldsymbol{x} = \boldsymbol{a}\alpha_1 + \boldsymbol{b}\alpha_2$ and $\boldsymbol{y} = \boldsymbol{c}\alpha_1 + \boldsymbol{d}\alpha_2$ in $\mathbb{F}_{q^2}^n$ where $\{\alpha_1,\ \alpha_2\}$ is a basis of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$ and the matrix

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_1^q & \alpha_2^q \end{pmatrix}$$

is such that $det(A) = \alpha_1\alpha_2^q - \alpha_1^q\alpha_2 \neq 0$.

Then,

$$\mathbf{x} \cdot \mathbf{y}^q = (\boldsymbol{a}\alpha_1 + \boldsymbol{b}\alpha_2) \cdot (\alpha_1^q\boldsymbol{c} + \alpha_2^q\boldsymbol{d}) = \alpha_1\alpha_1^q\boldsymbol{a}\cdot\boldsymbol{c} + \alpha_1\alpha_2^q\boldsymbol{a}\cdot\boldsymbol{d} + \alpha_2\alpha_1^q\boldsymbol{b}\cdot\boldsymbol{c} + \alpha_2\alpha_2^q\boldsymbol{b}\cdot\boldsymbol{d}$$

and

$$\mathbf{x}^q \cdot \mathbf{y} = (\boldsymbol{a}\alpha_1^q + \boldsymbol{b}\alpha_2^q) \cdot (\boldsymbol{c}\alpha_1 + \boldsymbol{d}\alpha_2) = \alpha_1\alpha_1^q\boldsymbol{a}\cdot\boldsymbol{c} + \alpha_1^q\alpha_2\boldsymbol{a}\cdot\boldsymbol{d} + \alpha_1\alpha_2^q\boldsymbol{b}\cdot\boldsymbol{c} + \alpha_2\alpha_2^q\boldsymbol{b}\cdot\boldsymbol{d}.$$

That is,

$$\frac{\mathbf{x}\cdot\mathbf{y^q} - \mathbf{x}^q\cdot\mathbf{y}}{\alpha_1\alpha_2^q - \alpha_1^q\alpha_2} = \frac{(\boldsymbol{a}\cdot\boldsymbol{d} - \boldsymbol{b}\cdot\boldsymbol{c})(\alpha_1\alpha_2^q - \alpha_1^q\alpha_2)}{\alpha_1\alpha_2^q - \alpha_1^q\alpha_2} = \boldsymbol{a}\cdot\boldsymbol{d} - \boldsymbol{b}\cdot\boldsymbol{c}.$$

We define,

$$\boldsymbol{x}\star\boldsymbol{y} \ = \ tr\left(\frac{\boldsymbol{x}\cdot\boldsymbol{y}^q - \boldsymbol{x}^q\cdot\boldsymbol{y}}{\alpha_1\alpha_2^q - \alpha_1^q\alpha_2}\right) = tr(\boldsymbol{a}\cdot\boldsymbol{d} - \boldsymbol{b}\cdot\boldsymbol{c}). \tag{4.2}$$

Then, $\mathbf{x}\star\mathbf{x} = 0$.

Since tr is linear over $\mathbb{F}_p$ and $-1 \in \mathbb{F}_p$ we get that $\mathbf{x}\star\mathbf{y} = -(\mathbf{y}\star\mathbf{x})$. For $r \in \mathbb{F}_p$, $(r\mathbf{x})\star\mathbf{y} = r(\mathbf{x}\star\mathbf{y})$ by linearity of tr and for $\mathbf{z} = \boldsymbol{v}\alpha_1 + \boldsymbol{w}\alpha_2 \in \mathbb{F}_{q^2}^n$ we have

$$(\mathbf{x}+\mathbf{z})\star\mathbf{y} = \mathbf{x}\star\mathbf{y} + \mathbf{z}\star\mathbf{x} \ \textit{ and } \ \mathbf{x}\star(\mathbf{y}\star\mathbf{z}) = \mathbf{x}\star\mathbf{y} + \mathbf{x}\star\mathbf{z}.$$

Therefore, "$\star$" is an alternating and bilinear form over $\mathbb{F}_p$, i.e., it is a **sympletic inner product**.

Observe that for $q = p$, a prime number,

$$\mathbf{x} \star \mathbf{y} = \boldsymbol{a} \cdot \boldsymbol{d} - \boldsymbol{b} \cdot \boldsymbol{c}$$

which corresponds, for $p = 2$, to Equation (3.14), the symplectic inner product obtained in [14].

Observe that $\mathbf{x} \star \mathbf{y} = -(\mathbf{x} *_a \mathbf{y})$, where "$*_a$" is given by Equation (2.17), see [31]. Therefore, given any basis of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$, $\mathbf{x} \perp_\star \mathbf{y}$ if and only if $\mathbf{x} \perp_{*_a} \mathbf{y}$, and according to Equation (3.5), if $\mathbb{F}_q$ has characteristic 2 and $\boldsymbol{x} \perp_H \boldsymbol{y}$, then $\mathbf{x} \perp_{*_a} \mathbf{y}$. That is, if $C \subset \mathbb{F}_{q^2}^n$ is an additive code such that $C \subset C^{\perp_H}$, then $C \subset C^{\perp_\star}$.

In particular, for a polynomial basis of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$, $\{1,\ \delta\}$, given $\mathbf{x} = \boldsymbol{a} + \delta\boldsymbol{b}$ and $\mathbf{y} = \boldsymbol{c} + \delta\boldsymbol{d}$ in $\mathbb{F}_{q^2}^n$

$$\mathbf{x} \star \mathbf{y} = tr\left(\frac{\mathbf{x} \cdot \mathbf{y}^q - \mathbf{x}^q \cdot \mathbf{y}}{\delta^q - \delta}\right).$$

Proposition 4.4. *Let $\mathbb{F}_q$ be a finite field of characteristic 2. Take two different linear codes over $\mathbb{F}_q$, C_0 and C_1, such that $C_0 \subset C_1$. Then, the additive code $\mathbf{GU}(C_0, C_1^\perp)$ yields an $[[n,\ k_1 - k_0,\ d]]_q$ quantum stabilizer code with $d = dist(\mathbf{GU}(C_1, C_0^\perp) \setminus \mathbf{GU}(C_0, C_1^\perp))$.*

Proof. From Lemma 3.1, $C = \mathbf{GU}(C_0, C_1^\perp)$ is an additive and nonlinear code over $\mathbb{F}_{q^2}$. From Theorem 3.4,

$$C \subset C^{\perp_{TH}} = \mathbf{GU}(C_1, C_0^\perp),$$

and from Equation (4.2)

$$C^{\perp_{TH}} \subset C^{\perp_\star}.$$

Observe that $|C| = q^{n-(k_1-k_0)}$. Then, we define the stabilizer group as $\mathbb{S} = \left\{E_{\boldsymbol{ab}} \in \xi_n \mid \boldsymbol{a} + \delta\boldsymbol{b} \in \mathbf{GU}\right.$ and we apply Theorem 2.17. $\square$

Lemma 4.1. *Let $C_0 \in \mathbb{F}_q^n$ be an Euclidean self-orthogonal code. Then, the trace-alternating duality for $C = \mathbf{GU}(2,\ C_0)$ is the same as Euclidean duality.*

Proof. If we take $C_0 \subset \mathbb{F}_q^n$ and $C_1 \subset \mathbb{F}_q^n$ such that $C_0 \subset C_1^{\perp}$, where $C_1^{\perp}$ is the Euclidean dual of C_1, then $C_0 + \delta C_1 \subset (C_0 + \delta C_1)^{\perp_\star} = (C_0 + \delta C_1)^{\perp_{*a}}$. Because in this case $\boldsymbol{a} \cdot \boldsymbol{d} = 0 = \boldsymbol{b} \cdot \boldsymbol{c}$. Therefore, $C = C_0 + \delta C_1$ is self-orthogonal with respect to the form $\star$. In particular, if $C_0 = C_1$, that is, C_0 is Euclidean-self-orthogonal, the linear code $\mathbf{GU}(2,\ C_0)$ is $\star-$self-orthogonal and

$$\mathbf{GU}(2,\ C_0)^{\perp_\star} = \mathbf{GU}(2,\ C_0^{\perp}) = \mathbf{GU}(2,\ C_0)^{\perp}.$$

This follows by taking $\mathbf{y} = \boldsymbol{c}\alpha_1 + \boldsymbol{d}\alpha_2 \in \mathbf{GU}(2,\ C_0^{\perp})$, for any $\mathbf{x} = \boldsymbol{a}\alpha_1 + \boldsymbol{b}\alpha_2 \in \mathbf{GU}(2,\ C_0)$,

$$\mathbf{x} \star \mathbf{y} = tr(\boldsymbol{a} \cdot \boldsymbol{d} - \boldsymbol{b} \cdot \boldsymbol{c}) = tr(0 - 0) = 0.$$

Thus,

$$\mathbf{GU}(2,\ C_0^{\perp}) \subset \mathbf{GU}(2,\ C_0)^{\perp_\star}.$$

But $|\mathbf{GU}(2,\ C_0^{\perp})| = q^{2n-2k_0} = |\mathbf{GU}(2,\ C_0)^{\perp_\star}|$. Then,

$$\mathbf{GU}(2,\ C_0)^{\perp_{*a}} = \mathbf{GU}(2,\ C_0)^{\perp_\star} = \mathbf{GU}(2,\ C_0)^{\perp}.$$

Therefore, we just need to establish classical Euclidean duality for our construction and work with classical self-orthogonal codes. □

Similar to *Theorem* 4.1, we can get

Theorem 4.2. *Let $C \subset \mathbb{F}_{q^2}^n$ be an $[n,\ k,\ d]_{q^2}$ Euclidean, i.e., classical self-orthogonal code, that is Frobenius invariant. Then, C yields an $[[n,\ n-2k,\ \geq d^{\perp}]]_q$ quantum stabilizer code that is pure to d.*

Proof. Let $C \subset \mathbb{F}_{q^2}^n$ be a linear code and C_0 its sub-field sub-code over $\mathbb{F}_q$. From Lemma 3.1,

$$\mathbf{GU}(2,\ C_0) = \{\boldsymbol{a} + \delta\boldsymbol{b} \mid \boldsymbol{a},\ \boldsymbol{b} \in C_0\}$$

is a linear code over $\mathbb{F}_{q^2}$. From Theorem 3.1 and taking $m = 2$, we get

$$C = \mathbf{GU}(2,\ C_0).$$

From Proposition 3.5 and Lemma 4.1, $C \subset C^{\perp} = \mathbf{GU}(2,\ C_0)^{\perp} = \mathbf{GU}(2,\ C_0^{\perp}) = C^{\perp_\star}$ and, applying *Theorem* 2.17, we obtain an $[[n,\ n-2k,\ \geq d^{\perp}]]_q$ quantum stabilizer code. □

Remark: *Theorem* 4.1 and *Theorem* 4.2 are important because, using them, we do not need to verify alternating or Hermitian self-orthogonality of the code C to construct quantum stabilizer codes, see *Corollary* 19 in [31], we just need to begin with classical self-orthogonal, i.e., Euclidean self-orthogonal codes. There are many classes of codes that are classically self-orthogonal.

4.3 OPEN PROBLEMS AND FUTURE DIRECTIONS

We have shown applications of our **GU** construction considering two linear codes, that is, we have studied the additive or linear code $\mathbf{GU}(C_0, C_1)$. Therefore, our first aim is to study the code $\mathbf{GU}(C_0, C_1, \cdots, C_{m-1})$ when $m \geq 3$. For example, we know from Lemma 3.1 that $\mathbf{GU}(C_0, C_1, \cdots, C_{m-1})$ is an additive nonlinear code when any two of the linear codes C_i, $0 \leq i \leq m-1$, are different. Then,

- Can we find an explicit formula for the weight distribution of such additive code when $m \geq 3$, as we found for $m = 2$?
- Can we find optimal additive or linear code when $m \geq 3$ as we got with $m = 2$?
- How to use our two-weight and three-weight codes with secret sharing schemes, along the lines of [2, 22, 46].?
- Construct strongly regular graphs from our two-weight codes.
- Establish whether our two-weight codes constructed via the generalized Simplex codes and **GU** construction are equivalent to the **RT1** codes of Calderbank and Kantor [16].

- When is the additive code $\mathbf{GU}(C_0, C_1, \cdots, C_{m-1})$ a few-weight code, given that the linear codes C_i have few weights, such as the simplex code or the first order Reed-Muller code.?
- If $m \geq 3$, under what conditions is the additive $\mathbf{GU}(C_0, C_1, \cdots, C_{m-1})$ code self-orthogonal.?

In *Chapter* 4 we gave an application of the **GU** codes to construct quantum stabilizer codes using Euclidean self-orthogonality. With respect to our quantum codes, we want to study quantum maximum distance separable codes ($QMDS$); that is, a quantum code $[[n,\ k,\ d]]_q$ satisfying

$$d = \frac{n-k}{2} + 1.$$

From [26], we have the following interesting questions:

- Can we construct $QMDS$ code $[[q^2+1,\ q^2+1-2d,\ d]]_q$ for q even?
- Are there $QMDS$ codes that are not related to classical MDS codes?
- Prove, disprove, or refine the $QMDS$ conjecture, which is: The length of any $QMDS$ code $[[n, k, d]]_q$ with $d \geq 3$ is bounded by $n \leq q^2 + 1$, with the exception of $[[q^2+2, q^2-4, 4]]_q$ for $q = 2^m$, when $n \leq q^2 + 2$.

We are also interested in:

- Construct **QECC** from separable and non-separable Goppa codes.

  In the thesis, I have partial results on the minimum distance. We would like to improve these results and also prove results on the dimension.
- Study Shor's factorization algorithm (1994). (Shor's algorithm could be used to break public-key cryptography schemes, such as the widely used RSA scheme.)

REFERENCES

[1] Salah A Aly, Andreas Klappenecker, and Pradeep Kiran Sarvepalli. On Quantum and Classical BCH Codes. *IEEE Transactions on Information Theory*, 53(3):1183–1188, 2007.

[2] Alexei Ashikhmin and Alexander Barg. Minimal vectors in linear codes. *IEEE Transactions on Information Theory*, 44(5):2010–2017, 1998.

[3] Alexei Ashikhmin and Emanuel Knill. Nonbinary Quantum Stabilizer Codes. *IEEE Transactions on Information Theory*, 47(7):3065–3072, 2001.

[4] E.F Assmus Jr and Harold F Mattson. Error-correcting codes: An axiomatic approach. *Information and Control*, 6(4):315–330, 1963.

[5] H. Barnum, C. Crepeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of Quantum Messages. *IEEE Press*, pages 449–458, 2002.

[6] Elwyn Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, Inc, 1968.

[7] J. Bierbrauer. *Introduction to CODING THEORY*. CRC Press Taylor Francis Group, second edition, 2017.

[8] Jürgen Bierbrauer, Daniele Bartoli, Giorgio Faina, Stefano Marcugini, and Fernanda Pambianco. The nonexistence of an additive quaternary [15, 5, 9]-code. *Finite Fields and Their Applications*, 36:29–40, 2015.

[9] Norman L Biggs. *Codes: An Introduction to Information Communication and Cryptography*. Springer, 2008.

[10] George Robert Blakley. Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on*, pages 313–313. IEEE Computer Society, 1979.

[11] Kenneth Bogart, Don Goldberg, and Jean Gordon. An Elementary Proof of the Macwilliams Theorem on Equivalence of Codes. *Information and Control*, 37(1):19–22, 1978.

[12] Joaquim Borges, Josep Rifa, and Victor A Zinoviev. On q-ary Linear Completely Regular Codes with $\rho = 2$ and Antipodal dual. *Advances in Mathematics of Communications*, 4(4):567, 2010.

[13] A Calderbank, P Shor, N Sloane, and E Rains. Quantum Error Correction and Orthogonal Geometry. *Physical Review Letters*, 78(3):405–408, 1997.

[14] A Robert Calderbank, Eric M Rains, PM Shor, and Neil JA Sloane. Quantum Error Correction Via Codes Over GF(4). *IEEE Transactions on Information Theory*, 44(4):1369–1387, 1998.

[15] A Robert Calderbank and Peter W Shor. Good Quantum Error-Correcting Codes Exist. *Physical Review A*, 54(2):1098–1106, 1996.

[16] Robert Calderbank and William M Kantor. The Geometry of Two-Weight Codes. *Bulletin of the London Mathematical Society*, 18(2):97–122, 1986.

[17] Dean Crnkovic, Andrea Svob, and Vladimir D Tonchev. Cyclotomic Trace Codes. *Algorithms*, 12(8):168, 2019.

[18] Ph Delsarte. *Two-weight linear codes and strongly regular graphs.* MBLE. Laboratoire de Recherches, 1971.

[19] Ph Delsarte. Weights of linear codes and strongly regular normed spaces. *Discrete Mathematics*, 3(1-3):47–64, 1972.

[20] Philippe Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Res. Rep. Suppl.*, 10:vi+–97, 1973.

[21] Cunsheng Ding, Jinquan Luo, and Harald Niederreiter. Two-weight Codes Punctured from Irreducible Cyclic Codes. In *Coding And Cryptology*, pages 119–124. World Scientific, 2008.

[22] Kelan Ding and Cunsheng Ding. A Class of Two-Weight and Three-Weight Codes and Their Applications in Secret Sharing. *IEEE Transactions on Information Theory*, 61(11):5835–5842, 2015.

[23] Ying Dong, Dan Hu, Sixia Yu, et al. Breeding quantum error-correcting codes. *Physical Review A*, 81(2):022322, 2010.

[24] Daniel Gottesman. Theory of Fault-Tolerant Quantum Computation. *Physical Review A*, 57(1):127–137, 1998.

[25] Daniel Eric Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997.

[26] Markus Grassl, M Rötteler, and T Beth. On quantum mds codes. In *Proc. Int. Symp. Inform. Theory, Chicago, USA*, page 356, 2020.

[27] K Hoffman and Kunze R. *Linear Algebra*. Prentice-Hall, Inc., second edition, 1971.

[28] W Cary Huffman and Vera Pless. *Fundamentals of Error-Correcting Codes*. Cambridge university press, 2010.

[29] Kenneth Ireland and Michael Rosen. *A classical introduction to modern number theory*, volume 84 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1990.

[30] Dieter Jungnickel and Vladimir D Tonchev. On bonisolis theorem and the block codes of Steiner triple systems. *Designs, Codes and Cryptography*, 86(3):449–462, 2018.

[31] Avanti Ketkar, Andreas Klappenecker, Santosh Kumar, and Pradeep Kiran Sarvepalli. Nonbinary Stabilizer Codes Over Finite Fields. *IEEE Transactions on Information Theory*, 52(11):4892–4914, 2006.

[32] Jon-Lark Kim and Vera Pless. Designs in additive codes over gf (4). *Designs, Codes and Cryptography*, 30(2):187–199, 2003.

[33] E Knill. Non-binary unitary error bases and quantum codes. Technical report, Los Alamos National Lab., NM (United States), 1996.

[34] Piyush P Kurur. Quantum Error Correcting Codes: An introduction. *Indian Institute of Technology Kanpur*, 2005.

[35] Giuliano G La Guardia. Constructions of New Families of Nonbinary Quantum Codes. *Physical Review. A*, 80(4), 2009.

[36] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and their Applications*. Cambridge university press, 1994.

[37] Wilde Mark M. *From Classical to Quantum Shannon Theory*. Cambridge University Press, 2019.

[38] Robert J McElliece. *The Theory of Information and Coding*. Addison-Wesley Publishing Company Inc, 1977.

[39] FJ McWilliams and NJA Sloane. *The Theory of Error-Correcting Codes*. North-Holland Amsterdam, 1977.

[40] Nicholas Patterson. The algebraic decoding of goppa codes. *IEEE Transactions on Information Theory*, 21(2):203–207, 1975.

[41] William Wesley Peterson and Edward J Weldon. *Error-Correcting Codes*. The Massachusetts Institute of Technology, second edition, 1972.

[42] Eric M Rains. Nonbinary Quantum Codes. *IEEE Transactions on Information Theory*, 45(6):1827–1832, 1999.

[43] Joseph M Renes. Quantum Information Theory. In Lecture Notes, 2015.

[44] Steven Roman. *Advanced Linear Algebra*. Springer, second edition, 2005.

[45] Benjamin Schumacher. Quantum Coding. *Physical Review A*, 51(4):2738, 1995.

[46] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[47] Peter W Shor. Fault-Tolerant Quantum Computation. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 56–65. IEEE, 1996.

[48] Andrew M Steane. Simple quantum error-correcting codes. *Physical Review A*, 54(6):4741, 1996.

[49] Douglas R. Stinson. An explication of secret sharing schemes. *Designs, Codes and Cryptography*, 2(4):357–390, 1992.

[50] W Trappe and L.C. Washinton. *Introduction to Cryptography with Coding Theory*. Pearson Prentice Hall, second edition, 2006.

[51] Zlatko Varbanov. Some new results for additive self-dual codes over gf (4). *Serdica Journal of Computing*, 1(2):213–227, 2007.

[52] Harold N Ward. An Introduction to Divisible Codes. *Designs, Codes and Cryptography*, 17(1):73–79, 1999.

[53] Harold N Ward and Jay A Wood. Characters and the Equivalence of Codes. *Journal of Combinatorial Theory, Series A*, 73(2):348–352, 1996.

[54] Edwin Weiss. Addendum: Linear codes of constant weight. *SIAM Journal on Applied Mathematics*, 15(1):229, 1967.

A GO-UP CONSTRUCTION AND APPLICATIONS

Eddie Arrieta Arrieta
eddie.arrieta@upr.edu
Departament of Mathematics
Chair: Heeralal Janwa Ph.D.
Grade: Doctorate in Mathematics
Date of Graduation:16 July 2021