THE ISLAMIC UNIVERSITY OF GAZA
DEANERY OF HIGHER STUDIES
FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICS

On Decoding Of Quadratic Residue Codes

PRESENTED BY
RAMEZ ZAHAR AL-SHORBASSI

SUPERVISED BY
Dr. MOHAMMED MAHMOUD AL-ASHKER

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT
FOR THE DEGREE OF MASTER OF MATHEMATICS

1431/2010

Abstract

A binary Quadratic Residue(QR) code of length n is an (n, (n+1)/2) cyclic code over $GF(2^m)$ with generator polynomial $g(x)$ where m is some integer. The length of this code is a prime number of the form $n = 8l + 1$ where l is some integer. The generator polynomial $g(x)$ is defined by

$$g(x) = \prod_{i \in Q_n} (x - \beta^i),$$

where β is a primitive n^{th} root of unity in the finite field $GF(2^m)$ with m being the smallest positive integer such that $n|2^m - 1$ and Q_n is the collection of all nonzero quadratic residues modulo n given by

$$Q_n = \{i | i \equiv j^2 \ mod \ n \ for \ 1 \leq j \leq n - 1\}.$$

Algebraic approaches to the decoding of the quadratic residue (QR) codes were studied in [2], [3], [4], [5], [6] and [13] . Here, in this thesis, some new more general properties are found for the syndromes of the subclass of binary QR codes of length $n = 8m + 1$ or $n = 8m - 1$.

A new algebraic decoding algorithm for the $(41, 21, 9)$ binary QR code is presented by having the unknown syndrome S_3 which is a necessary condition for decoding the $(41, 21, 9)$ QR code.

Keywords: cyclic codes, Newton identities, QR code.

Dedication

To the spirit of my father...

To my mother

To my wife

To my sons magd, and ody

To all knowledge seekers...

Acknowledgments

First of all, gratitude and thanks to **Almighty Allah** who always helps and guides me.

I would like to express my sincere appreciation and thanks to my supervisor Dr. Mohammed Mahmoud Al-ashker for his ceaseless help and supervision during the preparation of this project. Thanks are also due to the examiners committee Dr. Fayik EL-Naoqe, and Dr. Arwa Ashour.

I would like to express my sincere thanks to all the staff members of mathematics department and all my teachers who taught me to come to this stage of learning.

Contents

Introduction

The beginning of coding theory goes back to the middle of the last century with the work of Shannons, Hamming, Golay and others. Historically coding theory originated as the mathematical foundation for the transmission of messages over noisy channels. In fact a multitude of diverse applications have been discovered such as the minimization of noise from compact disc recordings the transmission of financial information a cross telephone lines, data transfer from one computer to another and so on. Coding theory deals with the problem of detecting and correcting transmission errors caused by noise on the channel. Mathematical background was at the beginning very little but with passing of time, various mathematical tools, such as group theory, ring theory, and linear programming have been applied to coding theory. Thus, coding theory has now become an active part of mathematical research. In many cases, the information to be sent is transmitted by a sequence of zeros and ones called binary codes, which means that the code is defined on the field $\{0, 1\}$.

The class of quadratic residue QR codes was introduced by Prange in 1958 [17]. It is a nice family of cyclic codes that has approximately $1/2$ code rates. There are many decoding algorithm was given for the binary QR of length $n = 8l \pm 1$, for some integer l In the past decades, most of decoding methods for the QR codes were used to solve the **Newton identities** that are nonlinear, multivariant equations of quite high degree. It becomes very difficult when the weight of the occurred error becomes large. Moreover, different QR codes use different sets of conditions to determine the **error locations**. As a result, it is not practical for software implementation.

In this thesis, A decoding algorithm was given for the (41, 21, 9) binary QR code. Some new more general properties are found for the syndromes of the subclass of binary QR codes of length $n = 8l+1$. Using these properties, the new theorems needed to decode this subclass of the QR codes are obtained and proved in [6]. For the binary (41, 21, 9) QR code, every known syndromes (resp., unknown syndromes) can be expressed as some power of S_1 (resp.,S_3). For the detail about syndromes we refer the reader to [13]. A new algebraic decoding algorithm for the $(41, 21, 9)$ binary QR code is presented.

The finite field to implement the $(41, 21, 9)$ QR code is $GF(2^{20})$. It contain 2^{20} elements. To calculate the syndrome polynomials over this field we use Maple 13. The algebraic method obtained in Section 4 for decoding the $(41, 21, 9)$ QR code is able to correct all of the four errors and a sufficient condition for the case of ν errors is found for this code. It is expected that some of the techniques developed here can be used to simplify the decoding of other QR codes.

The material of this thesis lies in four chapters.

Chapter 1:- Includes algebraic preliminaries, basic definitions of coding theory, encoding and decoding method, basic definitions on cyclic and quadratic residue codes.

Chapter 2:- Includes generator and parity check matrices of cyclic codes, encoding and decoding method of cyclic codes, the generating idempotent $e(x)$ for a cyclic code and Meggitt decoding method of cyclic codes.

Chapter 3:- Includes decoding binary quadratic residue codes of prime length as $(41, 21, 9)$, $(47, 24, 11)$ and $(73, 37, 13)$.

Chapter 4:- Includes decoding the binary Golay code and gives two algebraic decoding methods of the binary $(23, 12, 7)$.

Chapter 1

Preliminaries

This chapter is divided into four sections. In section one, we set some definitions which will be applied throughout the thesis. In section two, we study generating and parity check matrices. In section three , we study the processes of encoding and decoding in general. finally in section four, we study a basic definition of cyclic codes and quadratic residue codes (QR code) and some of their properties.

1.1 Basic Algebra

The purpose of this section is to review some basic facts that will be needed through the thesis.

Definition 1.1.1. [9] A nonempty set of elements G is said to form a **group** if in G there is defined a binary operation, called the product and denoted by (.) such that

1. $a, b \in G$ implies that $a.b \in G$ (closed).
2. $a, b, c \in G$ implies that $a.(b.c) = (a.b).c$ (associative).
3. There exists an element $e \in G$ such that $a.e = e.a = a$ for all $a \in G$(the existence of an identity element in G).

4. For every $a \in G$ there exists an element $a^{-1} \in G$ such that $a.a^{-1} = a^{-1}.a = e$ (the existence of inverse in G).

Definition 1.1.2. [9] A nonempty subset H of a group G is said to be a **subgroup** of G if, under the product in G, H itself forms a group.

Definition 1.1.3. [9] A nonempty set R is said to be an **associative ring** if in R, there are defined two operations, denoted by + (called addition) and . (called multiplication) respectively, such that for all a, b, c, in R :

1) $a + b$ is in R.

2) $a + b = b + a$

3) $(a + b) + c = a + (b + c)$.

4) There is a unique element 0 in R such that $a + 0 = a$ (for every a in R).

5) There exists a unique element $-a$ in R such that $a + (-a) = 0$.

6) $a \,.\, b$ in R.

7) $a \,.\, (\, b \,.\, c\,) \;=\; (\, a \,.\, b\,) \,.\, c$.

8) $a \,.\, (\, b \;+\; c\,) \;=\; a \,.\, b \;+\; a \,.\, c$ and $(\, b \;+\; c\,).\; a \;=\; b \,.\, a \;+\; c \,.\, a$ (the two distributive laws hold).

If the multiplication in R is such that $a.b = b.a$ for every a, b in R, then we call R a **commutative ring**

Definition 1.1.4. [9] A **ring homomorphism** ϕ from a ring R to a ring S is a mapping from R to S such that for all a, b in R:

1. $\phi(a + b) = \phi(a) + \phi(b)$.

2. $\phi(ab) = \phi(a)\phi(b)$.

Example 1.1.1. *[11] Let $R[x]$ denote the ring of all polynomials with real coefficients. The mapping $f(x) \longrightarrow f(1)$ is a ring homomorphism from $R[x]$ onto R.*

Definition 1.1.5. [9] A ring homomorphism that is both one-to-one and onto is called an **isomorphism**.

Definition 1.1.6. [19] Let R be a ring. A nonempty subset I of R is called an **ideal** if

1. both $a+b$ and $a-b$ belong to I, for all $a, b \in I$.

2. $r.a \in I$, for all $r \in R$ and $a \in I$.

Notation 1.1.1. *For any ring R , $\{0\}$ and R are ideals of R.*

Definition 1.1.7. [19] An ideal I of a ring R is called a **principle ideal** if it is generated by an element $g \in I$ such that $I = \langle g \rangle$, where

$$I = \langle g \rangle := \{gr : r \in R\} \quad .$$

A ring R is a **principal ideal ring** if every ideal of R is principal. The element g is called a **generator** of I and I is said to be generated by g.

Definition 1.1.8. [19] Let R be a ring and let A be ideal of R. The set of cosets $\{r + A | \ r \in R\}$ is called **factor ring**.

Example 1.1.2. *[23] In the ring $F_2[x]/(x^3-1)$, the subset $I = \{0, 1+x, x+x^2, 1+x^2\}$ is an ideal.*

Definition 1.1.9. [9] An ideal $U \neq R$ in a ring R is said to be a **maximal ideal** of R if whenever U is an ideal of R such that $M \subset U \subset R$, then either $R = U$ or $M = U$.

Example 1.1.3. *The maximal ideals of Z_{36} ,are $\langle 2 \rangle$ and $\langle 3 \rangle$.*

Example 1.1.4. *[23] In the ring $F_2[x]/(x^3-1)$, the subset*

$$I := \{0, 1+x, x+x^2, 1+x^2\},$$

is principal (i.e.$I = \langle 1+x \rangle$). Now, note that;

$$0.(1+x) = 1+x^3 = 0 = (1+x+x^2)(1+x)$$
$$1.(1+x) = 1+x = (x+x^2)(1+x)$$
$$x.(1+x) = x+x^2 = (1+x^2)(1+x)$$
$$x^2.(1+x) = 1+x^2 = (1+x)(1+x)$$

Definition 1.1.10. [9] A **zero-divisor** is a nonzero element a of a commutative ring R such that there is a nonzero element $b \in R$ with $ab = 0$.

Definition 1.1.11. [9] Two polynomials $f_1, f_2 \in R[x]$ are called **coprime** if $< f_1 > + < f_2 >= R[x]$, or equivalently , if there exist $g_1, g_2 \in R[x]$ such that $f_1 g_1 + f_2 g_2 = 1$. A polynomial $f \in R[x]$ is **called regular** if it is not a zero divisor.

Definition 1.1.12. [9] A polynomial $f(x) \in F[x]$ is said to be **irreducible** over a field F if whenever $f(x) = a(x)\ b(x)$ with $a(x),\ b(x) \in F[x]$ then one of $a(x)$ or $b(x)$ has degree 0(constant), otherwise $f(x)$ is **reducible** .

Example 1.1.5. *[11] The polynomial $f(x) = 2x^2 + 4$ is irreducible over Q*

Example 1.1.6. *[11] The polynomial $f(x) = x^2 + 1$ is irreducible over Z_3, but reducible aover Z_5.*

Definition 1.1.13. [19] We denote by $F_q[x]/(1 + x^n)$, the ring of all polynomial, modulo $(1 + x^n)$ over the field F_q.
A polynomials $I(x) \in R_n$ is called **idempotent**, if $I^2(x) = I(x)\ \ mod\ (1 + x^2)$.

Example 1.1.7. *[23] $e(x) = x^3 + x^5 + x^6 \in \mathcal{R}_7$ over $F_2[x] \Rightarrow e^2(x) = (x^3 + x^5 + x^6)^2 = x^3 + x^5 + x^6$*
$\therefore$ $e(x)$ is an idempotent in $\mathcal{R}_7$.

Definition 1.1.14. [19] A **field** is a nonempty set F of elements with two operations "+" and "." satisfying the following axioms. For all $a, b, c \in F$:

1. F is closed under + and . ; i.e, $a + b$ and $a.b$ are in F.
2. Commutative laws: $a + b = b + a,\ \ a.b = b.a$.
3. Associative:$(a + b) + c = a + (b + c),\ \ (a.b).c = a.(b.c)$.
4. Distributive law: $a.(b + c) = a.b + a.c$.

Furthermore, two distinct identity elements 0 and 1 must exist in F satisfying the following:

1. $a + 0 = a$ for all $a \in F$.

2. $a.1 = a$ and $a.0 = 0$ for all $a \in F$.

3. For any a in F, there exists an additive inverse element $(-a)$ in F such that $a + (-a) = 0$.

4. For any $a \neq 0$ in F, there exists a multiplicative inverse element a^{-1} in F such that $a.a^{-1} = 1$.

Remarks 1.1.1. [8]

1. If p is a prime we let $GF(p)$ (Galois field with p elements) denote the integer ring modulo p.

2. $\mathbb{Z}_p = \{0, 1, 2, \cdots, p-1\}$, group under + mod p, the order of $\mathbb{Z}_p$ is p.
   $\mathbb{Z}^* = \{1, 2, \cdots, p-1\} = \mathbb{Z}_p/\{0\}$ is cyclic group under multiplication mod p and order $\mathbb{Z}_p/\{0\} = p - 1$.
   $\mathbb{Z}_p = \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = GF(p)$.
   p is called the characteristic of $\mathbb{F}_p$.

3. **Fields of order $q = p^m$:** Let $\mathbb{F}_p$ be a field, the set $\mathbb{F}[x]$ of all polynomials over $\mathbb{F}_p$, is a P.I.D (principal ideal domain).

4. We denote this field by $\mathbb{F}_q$ where $q = p^m$ or $\mathbb{F}_q = GF(p^m)$ the Galois field of order $q = p^m$ which is the extension of the prime field $\mathbb{F}_p$. The characteristic of $\mathbb{F}_{p^m}$ is p. The field $\mathbb{F}_{p^m}$ is a vector space over $\mathbb{F}_p$ of dim m and $\mathbb{F}_p$ is a prime subfield of $\mathbb{F}_q$.

5. The set $\mathbb{F}_q^* = \mathbb{F}_q/\{0\}$ is the set of non zero elements in $\mathbb{F}_q$ and it is a multiplicative group.

Theorem 1.1.2. *[23]*

(i) The group $\mathbb{F}_q^*$ is cyclic of order $q-1$ under the multiplication of $\mathbb{F}_q$.

(ii) If γ is a generator of this cyclic group, then $\mathbb{F}_q = \{0, 1 = \gamma^0, \gamma, \gamma^2, \cdots, \gamma^{q-2}\}$ and $\gamma^i = 1$ if and only if $(q-1)|i$.

Definition 1.1.15. [8] A generator $\gamma \in \mathbb{F}_q^*$ is called **a primitive element** of $\mathbb{F}_q$.

Theorem 1.1.3. *[8] Let $\mathbb{F}_q$ be a finite field. For every $a \in \mathbb{F}_q$, $a^{|\mathbb{F}_q|} = a^q = a$.*

Remark 1.1.1. : If γ is a primitive element of $\mathbb{F}_q$. Then $\gamma^{q-1} = 1$, hence $(\gamma^i)^{q-1} = 1$ for $0 \leq i \leq q-1$, then any element of $\mathbb{F}_q^*$ is a root of the polynomial $x^{q-1} - 1 \in \mathbb{F}_q[x]$ and hence $x^q - x$, and 0 are a root of $x^q - x$, so all elements of $\mathbb{F}_q$ are roots of $x^q - x$.

Definition 1.1.16. [9] A nonempty set V is said to be a **vector space** over a field F if V is an abelian group under an operation which we denote by $(+)$, and if for every $\alpha \in F$, $v \in V$ there is defined an element, written αv subject to

1. $\alpha(v + w) = \alpha v + \alpha w$;
2. $(\alpha + \beta)v = \alpha v + \beta v$;
3. $\alpha(\beta v) = (\alpha\beta)v$;
4. $1v = v$;

   for all $\alpha, \beta \in F$, $v, w \in V$ where (the 1 represent the unit element of F under multiplication).

Definition 1.1.17. [9] If V is a vector space over F and if $W \subset V$, then W is a **subspace** of V whenever $w_1, w_2 \in W$, $\alpha, \beta \in F$ implies that $\alpha w_1 + \alpha w_2 \in W$.

Definition 1.1.18. [19] Let X be a vector space over F_q. A set of vectors $\{\mathbf{x_1}, ..., \mathbf{x_r}\}$ in X are **linearly independent** if

$$\lambda_1 \mathbf{x_1} + ... + \lambda_r \mathbf{x_r} = 0 \Rightarrow \lambda_1 = ... = \lambda_r = 0$$

Definition 1.1.19. [19] Let V be a vector space over F_q and let $S = \{v_1, v_2, ..., v_k\}$ be a nonempty subset of V. The **(linear) span** of S is defined as

$$< S >= \{\lambda_1 v_1 + ... + \lambda_k v_k : \lambda_i \in F_q\}.$$

Given a subspace C of V, a subset S of C is called spanning set of C if $C =< S >$.

Definition 1.1.20. [19] Let X be a vector space over F_q. A nonempty subset $B=\{\mathbf{x_1}, ..., \mathbf{x_r}\}$ of X is called a **basis** for X if $X =< B >$, B is a spanning set for X and linearly independent.

Definition 1.1.21. [19] A vector space X over a finite field F_q can have many basis; but all basis contain the same number of elements. This number is called the **dimension** of X over F_q, denoted by $dim(X)$.

1.2 Background on coding theory

In this section, we define alphabet, codes, codewords, codes over fields, Hamming distances, Hamming weights, linear code, generator matrix and parity check matrix .

We begin with some basic definitions. See [19].

⋆ **Strings and codes:** Let $C = \{c_1, c_2.....c_q\}$ be a finite set called alphabet. A **string** or a **word** over the alphabet A is any sequence of elements of A, we will usually (but not always)write words in the form $\mathbf{c} = c_{i_1} c_{i_2} ..... c_{i_k}$
using juxtaposition of symbols. The empty word 0 is the unique word with no symbols. The **length** of a word **c** denoted by **len(c)** is the number of the alphabet symbols appearing in the word. The set of all words (strings) over C will denoted by A^*.

⋆ **q-ary:**Let $C = \{c_1, c_2.....c_q\}$ be a finite set which we call a code alphabet. An **q-ary** is a nonempty subset A of the set C^* of all words over C. The size q of the code alphabet is called the **radix** of the code and the element of the code are called **codewords**.

The field $F_2 = \{0, 1\}$ has a very special place in history of coding theory, and codes over F_2 are called binary codes. Similarly, codes over $F_3 = \{0, 1, 2\}$ are termed ternary codes , while codes over $F_4 = \{0, 1, w, \overline{w}\}$ are called quaternary codes.The term "quaternary" has also been used to refer to codes over the ring $Z_4 = \{0, 1, 2, 3\}$ of integers modulo 4.

⋆**Fixed and variable length codes:** If all codewords in a code C have the same length we say that C is a **fixed length code**, or **block code**. If C contains codes of different lengths, we say that C is a **variable length code**. We will consider only **block** codes. We shall denote the number of codewords in a code C by $|C|$.
Let A^n be the set of all strings of length n. Any nonempty subset C of A^n is called a q-ary block code , each string in C is called codeword. If $C \subset A^n$ contains M codewords, it is customary to say that C has length n and size M, we denote this by (n, M)-code.

Example 1.2.1. *[23]*

The binary code $C = \{000, 100, 010, 001, 110, 101, 011, 111\}$ *contains* $M = |C| = 2^3 = 8$ *words.*

Fact:*For any binary code* C *of length* n, $1 \leq |C| \leq 2^n$.

For the purpose of this thesis, codes will have alphabet as a field or a ring under addition and multiplication. In fact, almost our codes' alphabet will be defined on **GF(q)**, a Galois field of q-elements and on commutative finite rings.

Definition 1.2.1. [8]

- The (**Hamming distance**) $d(x, y)$ between two vectors $x,\ y \in \mathbb{F}_q^n$ is defined to be the number of coordinates in which x and y differ.
- The (**Hamming weight**) $wt(x)$ of a vector $x \in \mathbb{F}_q^n$ is the number of nonzero coordinates in x.

Definition 1.2.2. [23] For a code C containing at least two words, the minimum distance of a code C, denoted by $d(C)$, is

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

Example 1.2.2. *[19] Let* $C = \{000000, 000111, 111222\}$ *be ternary code. Then* $d(c) = 3$ *since*
$d(000000, 000111) = 3$, $d(000000, 111222) = 6$, $d(000111, 111222) = 6$. *Hence, C is a ternary* $(6, 3, 3)$*-code.*

Remark 1.2.1. [19] For every element $\mathbf{x}$ of F_q, we can define the Hamming weight as follows:

$$wt(\mathbf{x}) = d(\mathbf{x}, 0) = \begin{cases} 1 & \text{if } \mathbf{x} \neq 0, \\ 0 & \text{if } \mathbf{x} = 0. \end{cases}$$

Theorem 1.2.1. *[23] If* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, *then* $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} - \mathbf{y})$. *If* $\mathcal{C}$ *is a linear code, then the minimum distance d is the same as the minimum weight of the nonzero codewords of* $\mathcal{C}$.

Proof. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{0}, \mathbf{y} - \mathbf{x}) = wt(\mathbf{y} - \mathbf{x})$ or $wt(\mathbf{x} - \mathbf{y})$ where $\mathbf{y} - \mathbf{x} \in \mathcal{C}$.
So the minimum distance $\{d(\mathbf{x}, \mathbf{y}),$ where $\mathbf{x} \neq \mathbf{y},\ \mathbf{x}, \mathbf{y} \in \mathcal{C}\} =$ the minimum weight $\{wt(\mathbf{x}-\mathbf{y}),$ where $\mathbf{x} \neq \mathbf{y},\ \mathbf{x}, \mathbf{y} \in \mathcal{C}\}$.
Then the minimum distance $d(\mathcal{C})$ = the minimum weight of the nonzero codewords of $\mathcal{C}$ = minimum$\{wt(\mathbf{a}) :\ \mathbf{a} \neq 0,\ \mathbf{a} \in \mathcal{C}\}$. □

Remark 1.2.2. [23] If the minimum distance of the $[n, k]-$ code $\mathcal{C}$ is d then the code will now be defined as $[n, k, d]$ code.

Definition 1.2.3. [23] If $\mathbf{x} = x_1x_2 \cdots x_n$ and $\mathbf{y} = y_1y_2 \cdots y_n$ are binary words then $\mathbf{x} \cap \mathbf{y} = (x_1y_1, x_2y_2, \cdots, x_ny_n)$. Thus $\mathbf{x} \cap \mathbf{y}$ has a 1 in the i^{th} position if and only if both $\mathbf{x},\ \mathbf{y}$ have a 1 in the i^{th} position.

Example 1.2.3. *If* $\boldsymbol{x}$*=(011011)*$\in F_2^6$ *and* $\boldsymbol{y}$*=(101011)*$\in F_2^6$, *then* $\mathbf{x} \cap \mathbf{y}$ *=(0, 0, 1, 0, 1, 1) .*

Theorem 1.2.2. *[23] The following hold:*

(i) *If* $\mathbf{x},\ \mathbf{y} \in \mathbb{F}_2^n$, *then* $wt(\mathbf{x}+\mathbf{y}) = wt(\mathbf{x}) + wt(\mathbf{y}) - 2wt(\mathbf{x} \cap \mathbf{y})$, *where* $\mathbf{x} \cap \mathbf{y}$ *is a vector in* $\mathbb{F}_2^n$, *which has 1's precisely in those positions where both* $\mathbf{x}$ *and* $\mathbf{y}$ *have 1's*

(ii) *If* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, *then* $wt(\mathbf{x} \cap \mathbf{y}) \equiv \mathbf{x} \cdot \mathbf{y}$ *mod* 2.

(iii) *If* $\mathbf{x} \in \mathbb{F}_2^n$, *then* $wt(\mathbf{x}) \equiv \mathbf{x} \cdot \mathbf{x}$ *mod* 2.

(iv) *If* $\mathbf{x} \in \mathbb{F}_3^n$, *then* $wt(\mathbf{x}) \equiv \mathbf{x} \cdot \mathbf{x}$ *mod* 3.

(v) *If* $\mathbf{x} \in \mathbb{F}_4^n$, *then* $wt(\mathbf{x}) \equiv \langle \mathbf{x} \cdot \mathbf{x} \rangle$ *mod* 2.

Proof.

(i) If $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, then $wt(\mathbf{x}+\mathbf{y}) = wt(\mathbf{x}-\mathbf{y}) = d(\mathbf{x}-\mathbf{y}, \mathbf{0})$
= the number of nonzero coordinates of $\mathbf{x}$+the number of nonzero coordinates of $\mathbf{y}-$ 2(the number of nonzero coordinates of $\mathbf{x} \cap \mathbf{y}) = wt(\mathbf{x}) + wt(\mathbf{y}) - 2wt(\mathbf{x} \cap \mathbf{y})$.

(ii) $wt(\mathbf{x} \cap \mathbf{y}) = wt(x_1y_1, x_2y_2, \cdots, x_ny_n)$ = the number of nonzero coordinates of $(\mathbf{x} \cap \mathbf{y}) \equiv (x_1y_1 + x_2y_2 + \cdots + x_ny_n)$ $(mod\ 2)$.

(iii) If $\mathbf{x} \in \mathbb{F}_2^n$, then $wt(\mathbf{x}) = \sum_{x_i \neq 0} x_i = \sum_{x_i \neq 0} x_i^2 \equiv \mathbf{x} \cdot \mathbf{x}$ $(mod\ 2)$.

(iv) If $\mathbf{x} \in \mathbb{F}_3^n$, then $wt(\mathbf{x}) = \sum_{x_i \neq 0} x_i = \sum_{x_i \neq 0} x_i^2 \equiv \mathbf{x} \cdot \mathbf{x}$ $(mod\ 3)$.

(v) If $\mathbf{x} \in \mathbb{F}_4^n$, then $wt(\mathbf{x}) = \sum_{x_i \neq 0} x_i = \sum_{x_i \neq 0} x_i^2 \equiv \mathbf{x} \cdot \bar{\mathbf{x}}$ $(mod\ 2) \equiv \langle \mathbf{x} \cdot \mathbf{x} \rangle$ $(mod\ 2)$.

□

Definition 1.2.4. [23] Let A_i or $A_i(\mathcal{C})$ be the number of codewords of weight i in $\mathcal{C}$. The list A_i for $0 \leq i \leq n$ is called the (**weight distribution**) weight distribution of $\mathcal{C}$.

Example 1.2.4. *[23] Let* $\mathcal{C}$ *be the binary code with generator matrix*

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

All codewords are

$$000000,\ 110000,\ 111100,\ 110011,\ 001111,\ 111111,\ 001100,\ 000011.$$

The weight distribution of $\mathcal{C}$ *are* $A_0 = 1,\ A_6 = 1,\ A_2 = 3,\ A_4 = 3$

Definition 1.2.5. [8] A code C is called a **linear code** if $\mathbf{x} + \mathbf{y}$ is a word in C whenever $\mathbf{x}$ and $\mathbf{y}$ are in C. That is, a linear code is a code which is closed under addition of words.

Note that all linear codes contain the **zero codewords**, denoted by $\mathbf{0} = 00.....0$. **Note** also that the dimension of a q-ary $[n, k]$ code is defined by $k = \log_{|F|} M$ where the size $M = q^k$ and the rate of C is $R = k/n$.

Example 1.2.5. $C = \{000, 111\}$ *is a linear code over* F_2, *since all four of the sums*

$$000 + 000 = 000$$
$$111 + 000 = 111$$
$$000 + 111 = 111$$
$$111 + 111 = 000$$

are in C. A linear code C must contain the zero codeword.

Definition 1.2.6. [19] A linear code C of length n over F_q is a subspace of ${F_q}^n$.

Definition 1.2.7. [23] A **generator matrix** for an $[n, k]$ code C is any $k \times n$ matrix G whose rows form a basis for C.

Remark 1.2.3. In general for any code C there are many generator matrices of size $k \times n$.

If C is any $[n, k]$-code, with generator matrix G, then the codewords in C are the linear combination of the rows of G.

Example 1.2.6. *[23]: Consider the binary code with generator matrix*

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

this matrix has three rows, then the dimension of the code is 3 and has 2^3 *codewords.*

Definition 1.2.8. [23] For any set of k independent columns of a generator matrix G, the corresponding set of coordinates form an **information set** of C, and the remaining $r = n - k$ coordinates are called the **redundancy set** of C.

Definition 1.2.9. [23] a generator matrix of the form

$$G = \left[\begin{array}{c|c} I_k & A \end{array} \right], \text{ where } I_k \text{ is the } k \times k \text{ identity matrix of size } k$$

is said to be in the **standard form**.

Definition 1.2.10. [23] **Parity check matrix** A parity check matrix for the $[n, k]$ code C, is an $(n - k) \times n$ matrix H such that

$$C = \{x \in F_q^n | Hx^T = 0\}.$$

Note that the rows of H will also be independent. In general, there are also several possible parity check matrices for C. The next theorem gives one of them when C has a generator matrix in standard form. In this theorem, A^T is the transpose of A.

Definition 1.2.11. [23] If C is a code over F_q then

$$C^{\perp} = \{x \in F_q^n : x \cdot c = 0 \text{ for all } c \in C\},$$

is the **dual code** of C.

Theorem 1.2.3. *[23] If $G = [I_k|A]$ is a generator matrix for the $[n, k]$code C in standard form, then $H = [-A^T|I_{n-k}]$ is a parity check matrix for C.*

Proof. we clearly have

$$HG^T = [-A^T|I_{n-k}] \left[\begin{array}{c} I_k \\ A^T \end{array} \right] = -A^T + A^T = 0.$$

Thus, C is contained in the kernel of the linear transformation $x \to Hx^T$. As H has rank $n - k$, this linear transformation has kernel of dimension k, which is also the dimension of C. The result follows.

□

Notation 1.2.4. *[23] Since* $GH^T = [I_k|A] \begin{bmatrix} -A \\ I_{n-k} \end{bmatrix}$ *= -A+A = 0. Hence the rows of H are orthogonal to the rows of G and since* $rank(H) = n - k = dim(C^{\perp})$*. We deduce that H is a generator matrix for the dual code* $C^{\perp}$*.*

Example 1.2.7. *:[23] The binary* $[7, 4]$*-code with generator matrix*

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right],$$

in standard form the first 4 coordinates form information set.

Example 1.2.8. *[23] The matrix*

$G = \left[\begin{array}{c|c} I_k & A \end{array}\right]$, *where* $G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$

is a generator matrix in standard form for a $[7, 4]$ *binary code that we denote by* $\mathcal{H}_3$*. The parity check matrix for* $\mathcal{H}_3$ *is*

$$H = \left[\begin{array}{c|c} A^{\top} & I_3 \end{array}\right] = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

This code $\mathcal{H}_3$ *is called the* $[7, 4]$ *Hamming code.*

Remark 1.2.4. any linear code has a generator matrix in **RREF**(Reduced Row Echelon Form).

Example 1.2.9. *To find the generator matrix G for the code* $C = \{0000, 1110, 0111, 1001\}$ *By elementary row operations we write*

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

so $G_1 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$ *is a generator matrix for* C*, also* $G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$,

is a generator matrix for C.

Note that G_2 *is in* ***RREF***

Definition 1.2.12. [9] Let $\mathbf{x} = x_1x_2\cdots x_n,\ \mathbf{y} = y_1y_2\cdots y_n \in F_q^n$ be two vectors, then the **inner product** is denoted by the formula $\mathbf{x}\cdot\mathbf{y} = \sum_{i=1}^{n} x_iy_i = x_1y_1 + \cdots + x_ny_n$.

1.3 Encoding and decoding

In this section we use generator matrix and parity check matrix to encode a message $\mathbf{x} = x_1x_2\cdots x_k$*, where* $G = [I_k|A],\ H = [-A^\top|I_{n-k}]$ *, and then we define syndrome and coset of a vector* $\mathbf{x}$ *and use them for decoding.*

Let $\mathcal{C}$ be an $[n,k]$ linear code over the field $\mathbb{F}_q$, with generator matrix G. This code has q^k codewords which will be in one to one correspondence with q^k messages. The simplest way to view these messages is as $k-$ tuples $\mathbf{x} \in \mathbb{F}_q^k$.

To encode the message $\mathbf{x}$ as a codeword $\mathbf{c} = \mathbf{x}G$, if $G = [I_k|A]$ in standard form, then the first k coordinates of the codeword $\mathcal{C}$ are the **information symbol x**; **the remaining** $n-k$ **symbols** are the parity check symbols, that is a redundancy added to $\mathbf{x}$ in order to help recover $\mathbf{x}$ if errors occur.

If G is not in standard form then there exists indices $i_1,\ i_2,\cdots,i_k$ such that the $k\times k$ matrix consisting of these k columns of G is the $k\times k$ identity matrix consisting of those k columns of G. Then the message is found in the $k-$ coordinates $i_1,\ i_2,\cdots,i_k$ of the codeword scrambled but otherwise unchanged, that is the message symbol x_j is in the component i_j of the codeword. This encoder is called systematic.

★ Encoding:

Let $\mathbf{x}$ be a message $\mathbf{x} = x_1x_2\cdots x_k$. Let $G = [I_k|A],\ H = [-A^\top|I_{n-k}]$. Suppose

$\mathbf{x} = x_1x_2\cdots x_k$ is encoded as a codeword $\mathbf{c} = c_1c_2\cdots c_n$ as G in standard form, $c_1c_2\cdots c_k = x_1x_2\cdots x_k$. So we need to determine the $n-k$ parity check symbols (redundancy symbols) $c_{k+1}c_{k+2}\cdots c_n$.

As $\mathbf{0} = H\mathcal{C}^\top = [-A^\top|I_{n-k}]\mathcal{C}^\top \implies \mathbf{0} = -A^\top\mathbf{x}^\top + I_{n-k}[c_{k+1}\cdots c_n]^\top \implies A^\top\mathbf{x}^\top = [c_{k+1}\cdots c_n]^\top$. See [23].

Example 1.3.1. *[23] Let G be the $[6,3,3]$ binary code with the following generator matrix and parity check matrix:*

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

Let $\mathbf{x} = x_1x_2x_3$ to obtain the codeword $\mathcal{C} = c_1c_2\cdots c_6$ using G to encode yields $\mathcal{C} = \mathbf{x}G = (x_1, x_2, x_3, x_1+x_2, x_2+x_3, x_1+x_3)$. Using H to encode $\mathbf{0} = H\mathcal{C}^\top$ leads to the system

$$\begin{aligned} 0 &= c_1 + c_2 + c_4 \\ 0 &= c_2 + c_3 + c_5 \\ 0 &= c_1 + c_3 + c_6 \end{aligned}$$

As G in standard form $c_1c_2c_3 = x_1x_2x_3$ and solving this system clearly gives the same codeword

$$\begin{aligned} c_4 &= c_1 + c_2 = x_1 + x_2 \\ c_5 &= c_2 + c_3 = x_2 + x_3 \\ c_6 &= c_1 + c_3 = x_1 + x_3 \end{aligned}$$

$\therefore\ \mathcal{C} = (x_1, x_2, x_3, x_1+x_2, x_2+x_3, x_1+x_3)$.

If G is not in standard form, since G has k independent rows, so there exists $n \times k$ matrix K such that $G\mathrm{K} = I_k$, K is called a right inverse for G and is not necessarily unique. As $\mathbf{c} = \mathbf{x}G \implies \mathbf{c}\mathrm{K} = \mathbf{x}G\mathrm{K} = \mathbf{x}I_k = \mathbf{x}$.

Theorem 1.3.1. *[7] A code C of distance d will at least detect all non-zero error patterns of weight less than or equal to $d-1$. Moreover, there is at least one error pattern of weight d which C will not detect.*

Example 1.3.2. *The code* $C = \{000, 111\}$, $d = 3$ *detects all error patterns of weight 1 or 2 and* C *does not detect the only error patterns of weight 3.*

Theorem 1.3.2. *[7] A code* C *of distance* d *will correct all error patterns of weight less than or equal to* $[(d-1)/2]$. *Moreover, there is at least one error pattern of weight* $1 + [(d-1)/2]$ *which* C *will not correct.*

Example 1.3.3. *The code* $C = \{000, 111\}$, $d = 3$ *correct all error patterns of weight 0 or 1, since* $(d-1)/2 = (3-1)/2 = 1$.

★ **Decoding:**

The decoding problem becomes one of finding an efficient algorithm that will correct up to t errors one of the most obvious decoding algorithm is to examine all codewords until one is found with distance t or less from the received vector. But this is efficient for codes of number of codewords.

Another obvious algorithm is to make a table consisting of a nearest codeword for each of the q^n vectors in $\mathbb{F}_q^n$ and then look up a received vector in the table in order to decode it. This is impractical if q^n is very large. See [23]

Definition 1.3.1. [11] **Coset of** C. If $C \subset F_q^n$ is a linear code (i.e subspace) the quotient space of F_q^n, modulo C is defined by

$$F_q^n/C = \{x + C | x \in F_q^n\}.$$

The set $x + C = \{x + c | c \in C\}$ is called a coset of C.

Note that $|x + C| = |C|$.

★ **Syndrome decoding for** $[n, k, d]$ **linear code** $\mathcal{C}$

We can devise an algorithm using a table with q^{n-k} rather than q^n entries where one can find the nearest codeword by looking up one of those q^{n-k} entries.

Notations 1.3.3.

- *The code $\mathcal{C}$ is an abelian subgroup of the additive group $\mathbb{F}_q^n$.*
- *If $\mathbf{x} \in \mathbb{F}_q^n$, then $\mathbf{x} + \mathcal{C}$ is a coset of $\mathcal{C}$.*
- *The cosets of $\mathcal{C}$ form a partition of $\mathbb{F}^n$ into q^{n-k} sets, each of size q^n.*
- *Two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ belong to the same coset if and only if $\mathbf{y} - \mathbf{x} \in \mathcal{C}$.*
- *The weight of a coset is the smallest weight of a vector in the coset.*
- *A coset leader is the vector in the coset of smallest weight.*
- *The zero vector is the unique coset leader of the code $\mathcal{C}$.*
- *In general every coset of weight at most $t = \lfloor \frac{d-1}{2} \rfloor$ has a unique coset leader.*

Definition 1.3.2. [23] Let H be the parity check matrix for $\mathcal{C}$. The **syndrome** of a vector $\mathbf{x}$ in $\mathbb{F}_q^n$ with respect to the parity check matrix H is the vector in $\mathbb{F}_q^{n-k}$ defined by $\text{syn}(\mathbf{x}) = H\mathbf{x}^\top$.

The code $\mathcal{C}$ consists of all vectors whose syndrome equal 0.
As rank $H = n - k$, every vector in $\mathbb{F}_q^{n-k}$ is a syndrome.

Theorem 1.3.4. *[23] Two vectors belong to the same coset if and only if they have the same syndrome.*

Proof. If $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{F}_q^n$ are in the same coset of $\mathcal{C}$, then $\mathbf{x}_1 - \mathbf{x}_2 = \mathbf{c} \in \mathcal{C} \Longrightarrow \mathbf{x}_1 = \mathbf{x}_2 + \mathbf{c}$. Therefore $\text{syn}(\mathbf{x}_1) = H(\mathbf{x}_2 + \mathbf{c})^\top = H\mathbf{x}_2^\top + H\mathbf{c}^\top = H\mathbf{x}_2^\top = \text{syn}(\mathbf{x}_2)$, then $\mathbf{x}_1, \mathbf{x}_2$ have the same syndrome and then lie on the same coset of $\mathcal{C}$.
If $\text{syn}(\mathbf{x}_1) = \text{syn}(\mathbf{x}_2) \Longrightarrow H(\mathbf{x_1} - \mathbf{x_2})^\top = 0 \Longrightarrow \mathbf{x_2} - \mathbf{x_1} \in \mathcal{C} \Longrightarrow \mathbf{x}_2 \in \mathbf{x}_1 + \mathcal{C}$. So $\mathbf{x_1}$, $\mathbf{x_2}$ lie on the same coset of $\mathcal{C}$. □

★ **Decoding method**

(i) Let $\mathbf{y}$ be a received vector, we seek an error vector $\mathbf{e}$ of smallest weight such that $\mathbf{c} = \mathbf{y} - \mathbf{e} \in \mathcal{C}$. We find the syndrome of (the coset of) the received vector $\mathbf{y} \in \mathbb{F}_q^n$. That is we compute $s = \text{syn}(\mathbf{y}) = H\mathbf{y}^\top$.

(ii) Finding a coset leader $\mathbf{e}$ in the coset of the received vector $\mathbf{y}$. Find a minimum weight vector $\mathbf{e} \in \mathbb{F}_q^n$ such that

$$s = \text{syn}(\mathbf{y}) = H(\mathbf{c} + \mathbf{e})^\top = H\mathrm{c}^\top + H\mathbf{e}^\top = H\mathbf{e}^\top.$$

(iii) Create a table pairing the syndrome with the coset leader, $\mathbf{y}$ is decoded as the code word $\mathbf{y} - \mathbf{e}$. The table is used to look up the syndrome and find the coset leader.

Remark 1.3.1. To find the syndrome $s = H\mathbf{e}^\top = \text{syn}(\mathbf{y})$ is equivalent to finding a smallest set of columns in H whose linear span contains the vector s.
The syndrome decoding for binary Hamming codes takes the form $[2^r - 1, 2^r - 1 - r, 3]$.
(i) After receiving a vector $\mathbf{y}$, compute its syndrome s using the parity check matrix H_r of the Hamming code $\mathcal{H}_r$.
(ii) If $s = 0$, then $\mathbf{y}$ is in the code and $\mathbf{y}$ is decoded as $\mathbf{y}$; otherwise, s is the binary numeral for some positive integer i and $\mathbf{y}$ is decoded as the codeword obtained from $\mathbf{y}$ by adding 1 to its ith bit. See [23].

Example 1.3.4. *[23] Construct the parity check matrix of the binary Hamming code $\mathcal{H}_4$ of length 15 where the columns are the binary numbers $1, 2, \cdots, 15$ in that order. Using this parity check matrix decode the following vectors, and then check that your decoded vectors are actually codewords.*

(a) $\mathbf{y}_1 = 001000001100100$

(b) $\mathbf{y}_2 = 101001110101100$.

Solution:

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

(a) Let $\mathbf{y}_1 = 001000001100100$ *Then*

$$H\mathbf{y}_1^\top = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = col_{13}.$$

So $\mathbf{e}_1 = 000000000000100$

$\therefore \mathbf{c}_1 = \mathbf{y}_1 - \mathbf{e}_1 = 001000001100000.$

(b)

$$H\mathbf{y}_2^\top = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = col_2$$

.So $\mathbf{e}_2 = 010000000000000$

$\therefore \mathbf{c}_2 = \mathbf{y}_2 - \mathbf{e}_2 = 111001110101100.$

Example 1.3.5. *[23] Let* $\mathcal{C}$ *be a linear* $[5, 2, 3]$ *code over* $\mathbb{F}_2$ *with generator matrix*

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

The cosets of the code $\mathcal{C}$ *are shown in the following table.*

00000	*10110*	*01011*	*11101*
00001	*10111*	*01010*	*11100*
00010	*10100*	*01001*	*11111*
00100	*10001*	*01111*	*11001*
01000	*11110*	*00011*	*10101*
10000	*00110*	*11011*	*01101*
00101	*10011*	*01110*	*11000*
10001	*00111*	*11010*	*01100*

Each row in the table is a coset of $\mathcal{C}$ *and the first vector in each row is the coset leader of minimum weight. The last two rows could start with any of the words* $00101, 11000, 10001,$ *or* 01100. *Suppose that the received word is* $\mathbf{y} = 01111$. *This word appears in the fourth row and the third column. The coset leader of the fourth row is* 00100, *and the decoded codeword is* 01011, *which is the first entry in the third column.*

We can use the syndrome decoding to decode the receive word $\mathbf{y}$ by using the parity check matrix

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

$H\mathbf{y}^\top = H(01111)^\top = (100)^\top = col_3$.

$\mathbf{e} = (00100)$ so $\mathbf{c} = \mathbf{y} - \mathbf{e} = (01111) - (00100) = (01011)$.

1.4 Basic definitions on cyclic and quadratic residue codes

In this section, we define cyclic code, duadic code and quadratic residue code

A linear code $\mathcal{C}$ of length n over $\mathbb{F}_q$ is **cyclic** provided that for each vector $\mathbf{c} = c_0c_1\cdots c_{n-2}c_{n-1}$ in $\mathcal{C}$ the vector $c_{n-1}c_0c_1\cdots c_{n-2}$ obtained from $\mathbf{c}$ by the cyclic shift of coordinate $i \longmapsto i+1 \pmod n$, is also in $\mathcal{C}$. So a **cyclic code** contains all n cyclic shifts of any codeword.

We will represent the codewords in a polynomial form. If $c_0c_1\cdots c_{n-2}c_{n-1} \in \mathbb{F}_q^n$ then $\mathbf{c}(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1} \in \mathbb{F}_q[x]$ of degree at most $n-1$. We order the terms of our polynomial from smallest to largest degree.
If $\mathbf{c}(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$, then $x\mathbf{c}(x) = c_{n-1}x^n + c_0x + c_1x^2 + \cdots + c_{n-2}x^{n-1}$ which would represent the codeword $\mathbf{c}$ cyclically shifted one to right, if $x^n = 1$.
Cylcic code $\mathcal{C}$ is invariant under a cyclic shift implies that if $\mathbf{c}(x)$ is in $\mathcal{C}$, then so is $x\mathbf{c}(x)$ provided we multiply modulo $x^n - 1$. See [23].

Remarks 1.4.1. [23]

(i) A linear code C is cyclic if and only if $\mathbf{c}(x) \in \mathcal{C} \Longrightarrow x \cdot \mathbf{c}(x) \pmod{x^n - 1} \in \mathcal{C}$. It follows that when $\mathbf{c}(x)$ is a codeword in a cyclic code, so are the words $x^i \cdot \mathbf{c}(x) \pmod{x^n - 1}$ for $i \geq 0$.

(ii) By linearity we conclude that in a cyclic code $\mathcal{C}$, $\mathbf{c}(x) \in \mathcal{C} \Longrightarrow \mathbf{u}(x)\mathbf{c}(x) \pmod{x^n - 1} \in \mathcal{C}$ for every $\mathbf{u}(x) \in \mathbb{F}_q[x]$, hence $\mathcal{C}$ is an ideal in $\mathcal{R}_n = \mathbb{F}_q[x]/(x^n - 1)$.

$\star$ **Generating idempotent of the cyclic code** [23] An element e of a ring satisfying $e^2 = e$ is called an **idempotent**, the ring $\mathcal{R}_n$ is **semi-simple** when $gcd(n, d) = 1$. therefore it follows from the Wedderburn Structure Theorems that each cyclic code in $\mathcal{R}_n$ contains a unique idempotent which generates the ideal. See [8], This idempotent is called the **generating idempotent of the cyclic code**.

Example 1.4.1. *[23] The generating idempotent for the zero cyclic code 0 is 0, while that for the cyclic code $\mathcal{R}_n$ is 1.*

Theorem 1.4.1. *[23] Let $\mathcal{C}$ be a cyclic code in $\mathcal{R}_n$. Then:*

(i) There exists a unique idempotent $e(x) \in \mathcal{C}$ such that $\mathcal{C} = \langle e(x) \rangle$, and

(ii) If $e(x)$ is a nonzero idempotent in $\mathcal{C}$, then $\mathcal{C} = \langle e(x) \rangle$ if and only if $e(x)$ is a unity of $\mathcal{C}$

Proof. See [23] □

Note that the way to find the generating idempotent e(x) for a cyclic code $\mathcal{C}$ from the generating polynomial $g(x)$ is to solve $a(x)g(x) + b(x)h(x) = 1$ for a(x) using the **Euclidean Algorithm**, where $h(x) = \frac{(x^n-1)}{g(x)}$. then reducing $a(x)g(x)$ modulo $x^n - 1$ produces $e(x)$.

Corollary 1.4.2. *[23] Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be cyclic codes over $\mathbb{F}_q$ with generator polynomials $g_1(x)$ and $g_2(x)$, respectively. then $\mathcal{C}_1 \subseteq \mathcal{C}_2$ if and only if $g_2(x) \mid g_1(x)$.*

Theorem 1.4.3. *[23] Let $\mathcal{C}$ be a cyclic code over $\mathbb{F}_q$ with generating idempotent $e(x)$. Then the generating polynomial of $\mathcal{C}$ is $g(x) = gcd(e(x), x^n - 1)$ computed in $\mathbb{F}_q[x]$.*

Proof. See [23] □

The previous theorem shows that We can produce $g(x)$ if we know $e(x)$.

Theorem 1.4.4. *[23] Let $\mathcal{C}_i$ be a cyclic code of length n over $\mathbb{F}_q[x]$ with generator polynomial $g_i(x)$ and generating idempotent $e_i(x)$ for i = 1 and 2. Then:*

(i) $\mathcal{C}_1 \cap \mathcal{C}_2$ has generator polynomial $lcm(g_1(x), g_2(x))$ and generating idempotent $e_1(x)e_2(x)$, and

(ii) $\mathcal{C}_1 + \mathcal{C}_2$ has generator polynomial $gcd(g_1(x), g_2(x))$ and generating idempotent $e_1(x) + e_2(x) - e_1(x)e_2(x)$.

Proof. (i) Let $l(x) = lcm(g_1(x), g_2(x)) \Longrightarrow g_1(x)|l(x),\ g_2(x)|l(x) \Longrightarrow l(x) = g_1(x)k_1(x),\ l(x) = g_2(x)k_2(x) \Longrightarrow e(x) \in \mathcal{C}_1$ and $l(x) \in \mathcal{C}_2$, since $\mathcal{C}_1 \cap \mathcal{C}_2$ is cyclic, then $\langle l(x)\rangle \subseteq \mathcal{C}_1 \cap \mathcal{C}_2$....(1)
Let $l'(x) \in \mathcal{C}_1 \cap \mathcal{C}_2 \Longrightarrow l'(x) \in \mathcal{C}_1,\ l'(x) \in \mathcal{C}_2 \Longrightarrow l'(x) = g_1(x)f_1(x),\ l'(x) = g_2(x)f_2(x) \Longrightarrow g_1|l'$ and $g_2(x)|l'(x) \Longrightarrow l'$ is a common multiple $\Longrightarrow l(x)|l'(x) \Longrightarrow \langle l'(x)\rangle \subseteq \langle l(x)\rangle \Longrightarrow \mathcal{C}_1 \cap \mathcal{C}_2 \subseteq \langle l(x)\rangle$....(2)
From (1) and (2) we have $\mathcal{C}_1 \cap \mathcal{C}_2 = lcm(g_1(x), g_2(x))$, since $g_1(x)|x^n-1$ and $g_2(x)|x^n-1 \Longrightarrow l(x)|x^n - 1 \Longrightarrow l(x)$ is a monic polynomial and generates $\mathcal{C}_1 \cap \mathcal{C}_2$.
If $\mathbf{c}(x) \in \mathcal{C}_1 \cap \mathcal{C}_2 \Longrightarrow \mathbf{c}(x) \in \mathcal{C}_1,\ \mathbf{c}(x) \in \mathcal{C}_2$ and $\mathbf{c}(x)(e_1(x)e_2(x) = \mathbf{c}(x)e_2(x) = \mathbf{c}(x) \Longrightarrow$

$e_1e_2 \in \mathcal{C}_1 \cap \mathcal{C}_2$ and e_1e_2 is the unity generator idempotent of $\mathcal{C}_1 \cap \mathcal{C}_2$.
(ii) Let $g(x) = \gcd(g_1(x), g_2(x))$ It follows from the **Euclidean Algorithm** that $g(x) = g_1(x)a(x) + g_2(x)b(x)$ for some $a(x)$ and $b(x)$ in $\mathbb{F}_q[x]$. So $g_1(x) \in \mathcal{C}_1 + \mathcal{C}_2$. Since $\mathcal{C}_1 + \mathcal{C}_2$ is cyclic, $\langle g(x)\rangle \subseteq \mathcal{C}_1 + \mathcal{C}_2$. On other hand $g(x)|g_1(x)$, which shows that $\mathcal{C}_1 \subseteq \langle g(x)\rangle$ by corollary 1.4.2, similarly $\mathcal{C}_2 \subseteq \langle g(x)\rangle$ implying that $\mathcal{C}_1 + \mathcal{C}_2 \subseteq \langle g(x)\rangle$. So $\mathcal{C}_1 + \mathcal{C}_2 = \langle g(x)\rangle$. Since $g(x)|x^n - 1$ as $g(x)|g_1(x)$ and $g(x)$ is monic, $g(x)$ is the generator polynomial for $\mathcal{C}_1 + \mathcal{C}_2 by corollary 1.4.2$. If $\mathbf{c}(x) = \mathbf{c}_1(x) + \mathbf{c}_2(x)$ where $\mathbf{c}_i(x) \in \mathcal{C}_i$ for $i = 1$ and 2, then $\mathbf{c}(x)(e_1(x) + e_2(x) - e_1(x)e_2(x)) = \mathbf{c}_1(x) + \mathbf{c}_1(x)e_2(x) - \mathbf{c}_1(x)e_2(x) + \mathbf{c}_2(x)e_1(x) + \mathbf{c}_2(x) - \mathbf{c}_2(x)e_1(x) = \mathbf{c}(x)$.
Thus (ii) follows, since $e_1(x) + e_2(x) - e_1(x)e_2(x) \in \mathcal{C}_1 + \mathcal{C}_2$ and $(e_1 + e_2 - e_1e_2)^2 = e_1 + e_2 - e_1e_2$. □

⋆ **Duadic codes:** [23] to define duadic codes we need some preliminary notation and result before. Let $\mathbb{Z}_p$ be the ring of integers modulo n, and let ξ_n denote the sub-code of even like vector in $\mathcal{R}_n = \mathbb{F}_q[x]/(x^n - 1)$. The code ξ_n is an [n, n-1] cyclic code whose dual code $\xi_n^{\perp}$ is the repetition code of length n which has generating idempotent

$$\overline{j}(x) = \frac{1}{n}(1 + x + x^2 + \cdots + x^{n-1})$$

so ξ_n has generating idempotent $1 - \overline{j}(x)\mu_{-1} = 1 - \overline{j}(x)$. Where μ_a is called a **multiplier** denote a permutation which maps idempotents of $\mathcal{R}_n$ to idempotents of $\mathcal{R}_n$, so μ_a is a function defined on $\{0, 1, \cdots, n-1\}$ by $i\mu_a = ia$ (mod n), where $gcd(a, n) = 1$ and a is some integer. In defining the duadic code, we will obtain two pairs of codes; one pair will be two **even-like codes**, which are thus sub-codes of ξ_n, and the other pair will be **odd-like codes**. It will be important to be able to tell when either a vector or a cyclic code in $\mathcal{R}_n$ is even-like or odd-like.

Lemma 1.4.5. *[23] Let $a(x) = \sum_{i=1}^{n-1} a_i x^i \in \mathcal{R}_n$. Also let $\mathcal{C}$ be a cyclic code in $\mathcal{R}_n$ with generating polynomial $g(x)$. Then:*

(i) $a(x)$ is even-like if and only if $a(1) = 0$ if and only if $a(x)\overline{j}(x) = 0$,

(ii) a(x) is odd-like if and only if $a(1) \neq 0$ *if and only if* $a(x)\overline{j}(x) = \alpha\overline{j}(x)$ *for some nonzero* $\alpha \in \mathbb{F}_q$,

(iii) $\mathcal{C}$ *is even-like if and only if* $g(1) = 0$ *if and only if* $\overline{j}(x) \notin \mathcal{C}$, *and*

(iv) $\mathcal{C}$ *is odd-like if and only if* $g(1) \neq 0$ *if and only if* $\overline{j}(x) \in \mathcal{C}$

Proof. See [23] □

Duadic codes come in two pairs, one even-like pair, which usually denoted by $\mathcal{C}_1$ and $\mathcal{C}_2$, and one odd-like pair, which usually denoted by $\mathcal{D}_1$ and $\mathcal{D}_2$. Let $e_1(x)$ and $e_2(x)$ be two even-like idempotents with $\mathcal{C}_1 = \langle e_1(x) \rangle$ and $\mathcal{C}_2 = \langle e_2(x) \rangle$. The codes $\mathcal{C}_1$ and $\mathcal{C}_2$ form a pair of even-like duadic codes provided the following:

$$e_1(x) + e_2(x) = 1 - \overline{j}(x)$$

, and there is a multiplier μ_a such that

$$\mathcal{C}_1\mu_a = \mathcal{C}_2 \quad and \quad \mathcal{C}_2\mu_a = \mathcal{C}_1$$

Note that we can replace $\mathcal{C}_1$ by $e_1(x)$ and $\mathcal{C}_2$ by $e_2(x)$, associate to $\mathcal{C}_1$ and $\mathcal{C}_2$ is the pair of odd-like duadic codes $\mathcal{D}_1 = \langle 1 - e_2(x) \rangle$ and $\mathcal{D}_2 = \langle 1 - e_1(x) \rangle$, where $1 - e_i(x) = 1$ for i = 1, 2. See [23].

Definition 1.4.1. [23] **square modulo n:** Let p be an odd prime and let a be in $\mathbb{Z}_P$ with $a1 \equiv 0$ (mod p). then a is a square if and only if $a^{\frac{p-1}{2}} \equiv 1$ (mod p).

Definition 1.4.2. [23] **Quadratic residue codes (QR codes)**: Quadratic residue codes are duadic codes over $\mathbb{F}_q$ of odd prime length $n = p$, which must be a square modulo n not dividing q, where q is a prime power that is a square modulo p.

Let $\mathcal{Q}_p$ denote the set of nonzero squares modulo p, and let $\mathcal{N}_p$ be the set of nonsquares modulo p. The sets $\mathcal{Q}_p$ and $\mathcal{N}_p$ are called the nonzero quadratic residues and quadratic residues modulo p, respectively.

Lemma 1.4.6. *[23] let p be an odd prime. The following holds:*

(i) $\mid \mathcal{Q}_p \mid = \mid \mathcal{N}_p \mid = \frac{p-1}{2}$.

(ii) Modulo p, we have $\mathcal{Q}_p a = \mathcal{Q}_p$, $\mathcal{N}_p a = \mathcal{N}_p$, $\mathcal{Q}_p b = \mathcal{N}_p$, and $\mathcal{N}_p b = \mathcal{Q}_p$ when $a \in \mathcal{Q}_p$ and $b \in \mathcal{N}_p$.

Proof. See [23] □

Theorem 1.4.7. *[23] Quadratic residue codes of odd prime length p exist over $\mathbb{F}_q$ if and only if $q \in \mathcal{Q}_p$.*

Theorem 1.4.8. *[23] Let $\mathcal{C}$ be a cyclic code of odd prime length p over $\mathbb{F}_q$, where q is a square modulo p. Let $e(x)$ be the generating idempotent of $\mathcal{C}$. The following hold:*

(i) $\mathcal{C}$ is a quadratic residue code or one of the trivial codes if and only if $e(x)\mu_c = e(x)$ for all $c \in \mathcal{Q}_p$.

(ii) If $\mathcal{C}$ is a quadratic residue code with generating idempotent $e(x)$, then

$$e(x) = a_0 + a_1 \sum_{i \in \mathcal{Q}_p} x^i + a_2 \sum_{i \in \mathcal{N}_p} x^i$$

, for some a_0, a_1, a_2 in $\mathbb{F}_q$.

(iii) If $c \in \mathcal{Q}_p$ and $\mathcal{C}$ is a quadratic residue code, then $\mu_c \in PAut(\mathcal{C})$.

Proof. See [23] □

Theorem 1.4.9. *[23] Let $\mathcal{C}$ be an even-like quadratic residue code of prime length p over $\mathbb{F}_q$ with idempotent $e(x)$. The following hold:*

(i) The four quadratic residue codes over $\mathbb{F}_q$ or any extension field of $\mathbb{F}_q$ have generating idempotents $e(x)$, $e(x)\mu_b$, $e(x) + \overline{j}(x)$, and $e(x)\mu_b + \overline{j}(x)$ for any $b \in \mathcal{N}_p$.

(ii) $e(x) + e(x)\mu_b = 1 - \overline{j}(x)$ for $b \in \mathcal{N}_p$.

(iii) The four quadratic residue codes over $\mathbb{F}_q$ have the same minimum weight and the same minimum weight codewords, up to scalar multiplication, as they do over an extension field of $\mathbb{F}_q$.

Proof. See [23] □

As in previous theorems we can find the generating idempotents of all QR codes over any field of characteristic 2 and 3 , see [23] page 238 - 244.

Chapter 2

Decoding of cyclic codes

This chapter is divided into four sections. In section one, we study generator and parity check matrices of cyclic codes. In section two we introduce many procedures for encoding and decoding cyclic codes. In section three, we give a method to Compute the generating idempotent e(x) for a cyclic code. Finally The first version of Meggitt decoding algorithm for cyclic codes is studied in section 4.

2.1 Generator and parity check matrices of cyclic codes

In this section we need some definitions and remarks to get a generator and parity check matrices of cyclic codes

Definition 2.1.1. [23] **The order of** q **modulo** n denoted by $\mathrm{ord}_n(q)$: is the smallest positive integer a such that $q^a \equiv 1(\mathrm{mod}\ n)$. That is if $q^a \equiv 1(\mathrm{mod}\ n) \Longrightarrow \mathrm{ord}_n(q) = a$.

If $t = \mathrm{ord}_n(q)$, then $\mathbb{F}_{q^t}$ contains a primitive n^{th} root of unity α, but no smaller extension field of $\mathbb{F}_q$ contains such a primitive root.
As α^i are distinct for $0 \leq i < n$ and $(\alpha^i)^n = 1, \mathbb{F}_{q^t}$ contains all the roots of $x^n - 1$. So $\mathbb{F}_{q^t}$ is called a splitting field of $x^n - 1$ over $\mathbb{F}_q$. So the irreducible factors of $x^n - 1$ over $\mathbb{F}_q$ must be the product of the distinct minimal polynomials of the n^{th} roots of unity in $\mathbb{F}_q$.

Suppose γ is a primitive element of $\mathbb{F}_{q^t}$. Then $\alpha = \gamma^d$ is a primitive nth root of unity where $d = \frac{q^t-1}{n}$. The roots of $M_\alpha(x)$ are $\{\gamma^{ds}, \gamma^{dsq}, \gamma^{dsq^2}, \cdots, \gamma^{dsq^{r-1}}\} = \{\alpha^s, \alpha^{sq}, \alpha^{sq^2}, \cdots, \alpha^{sq^{r-1}}\}$, where r is the smallest positive integer such that $dsq^r \equiv ds (\text{mod } q^t - 1)$. But $dsq^r \equiv ds (\text{mod } q^t - 1)$ if and only if $sq^r \equiv s (\text{mod } n)$, because $(dsq^r \equiv ds (\text{mod } dn) \Longrightarrow sq^r \equiv s (\text{mod } n))$.

This leads us to extend the notion of q-cyclotomic cosets.

Definition 2.1.2. [23] Let s be an integer with $0 \leq s < n$. The **q-cyclotomic cosets** of s modulo n is the set $\mathrm{C}_s = \{s, sq, \cdots, sq^{r-1}\} \pmod n$, where r is the smallest positive integer such that $sq^r \equiv s \pmod n$.

Definition 2.1.3. Monic Polynomial: A polynomial $x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ in which the coefficient of the highest order term is 1.

Definition 2.1.4. [23] If $\mathbb{E}$ is the extension field $\mathbb{F}_q$, then $\mathbb{E}$ is a vector space over $\mathbb{F}_q$, and $\mathbb{E} = \mathbb{F}_{q^t}$ for some positive t. Each element $\alpha \in \mathbb{E}$ is a root of the polynomial $x^{q^t} - x$. Thus there is a **monic polynomial** $M_\alpha(x)$ in $\mathbb{F}_q[x]$ of smallest degree which has α as a root, this polynomial is called the minimal polynomial of α over $\mathbb{F}_q$.

Note that C_s is the orbit of the permutation $i \longmapsto iq \pmod n$ that contains s. The distinct $q-$ cyclotomic cosets modulo n partition the set of integers $\{0, 1, 2, \cdots, n-1\}$ into disjoint sets.

Theorem 2.1.1. *[23] Let n be a positive integer relatively prime to q. Let $t = ord_n(q)$. Let α be a primitive n^{th} root of unity in $\mathbb{F}_{q^t}$.*

*(i) For each integer s with $0 \leq s < n$, the **minimal polynomial** of α^s over $\mathbb{F}_q$ is $\mathrm{M}_{\alpha^s}(x) = \prod_{i \in \mathrm{C}_s}(x - \alpha^i)$, where C_s is the q-cyclotomic cosets of s modulo n.*

(ii) The conjugates of α^s are the elements α^i with $i \in \mathrm{C}_s$.

(iii) Furthermore $x^n - 1 = \prod_s \mathrm{M}_{\alpha^s}(x)$ in the factorization of $x^n - 1$ into irreducible factors over $\mathbb{F}_q$, where s runs through a set of representation of the q-cyclotomic cosets modulo n.

Example 2.1.1. *[23] Consider the polynomial* $x^9 - 1$ *over* $\mathbb{F}_2$. *Since* $q = 2$ *and* $n = 9$, *then the* $2-$ *cyclotomic cosets over* 9 *are* $\mathrm{C}_0 = \{0\},\ \mathrm{C}_1 = \{1, 2, 4, 8, 7, 5\},\ \mathrm{C}_3 = \{3, 6\}$. *So* $ord_n(2) = 6$ *and the primitive ninth root of unity lie in* $\mathbb{F}_{64}$ *but no smaller extension field of* $\mathbb{F}_2$. *Hence* $x^9 - 1$ *factors into an irreducible factors as*

$$x^9 - 1 = (x^3)^3 - 1 = (x^3 + 1)(x^6 + x^3 + 1) = (x - 1)(x^2 + x + 1)(x^6 + x^3 + 1).$$

The polynomial $\mathrm{m}_{\alpha^0}(x) = \mathrm{M}_1(x) = x + 1,\ \mathrm{M}_\alpha(x) = x^6 + x^3 + 1,\ \mathrm{M}_{\alpha^3}(x) = x^2 + x + 1$, *where* α *is a primitive ninth root of unity in* $\mathbb{F}_{64}$. *The only irreducible polynomial of degree 2 over* $\mathbb{F}_2$ *is* $x^2 + x + 1 = \mathrm{M}_{\alpha^3}(x)$ *(*α^3 *is a primitive third root of unity in* $\mathbb{F}_{64}$.

Definition 2.1.5. [23] The order $\mathrm{ord}_n(q)$ of q modulo n is the smallest positive integer a such that $q^a \equiv 1 (\mathrm{mod}\ n)$. That is if $q^a \equiv 1 (\mathrm{mod}\ n) \Longrightarrow \mathrm{ord}_n(q) = a$.

Theorem 2.1.2. *[23] The size of each q- cyclotomic coset is a divisor of* $ord_n(q)$. *Furthermore the size of* C_1 *is* $ord_n(q)$.

Proof. Let $t = \mathrm{ord}_n(q)$ and let m be the size of C_s. Then $M_{\alpha^s}(x)$ has degree m where α is a primitive n^{th} root of unity, so $m|t$ and hence there exists a subfield $\mathbb{F}_{q^m} = \mathbb{F}_{p^{mr}}$ of $\mathbb{F}_{q^t} = \mathbb{F}_{p^{mt}} \Longrightarrow mr|mt \Longrightarrow m|t$ see [23] theorem 3.7.3. The fact that the size of C_1 is $\mathrm{ord}_n(q)$ follows directly from the definitions of $q-$ cyclotomic cosets and $\mathrm{ord}_n(q)$ as mentioned prior to definition 2.1.5 , because $\mathrm{C}_1 = \{1, q, q^2, \cdots, q^{r-1}\} \ (\mathrm{mod}\ q^t - 1)$, where r is the smallest positive integer such that $q^r \equiv 1 \ (\mathrm{mod}\ q^t - 1) \Longrightarrow r = t$. □

Notations 2.1.3. *cite5The cyclic codes of length n over* $\mathbb{F}_q$ *are precisely the ideals of* $\mathcal{R}_n = \mathbb{F}_q[x]/(x^n - 1)$, $\mathbb{F}_q[x]$ *is principal ideal domain also the ring* $\mathcal{R}_n$ *is principal, hence cyclic codes are the principal ideal of* $\mathcal{R}_n$, *when writing a codeword of a cyclic code as* $\mathbf{c}(x)$, *we technically mean the coset* $\mathbf{c}(x) + (x^n - 1)$ *in* $\mathcal{R}_n$. *We think the elements of* $\mathcal{R}_n$ *as the polynomial in* $\mathbb{F}_q[x]$ *of degree less than n with multiplication being carried out modulo* $x^n - 1$.

Theorem 2.1.4. *[23] Let* $\mathcal{C}$ *be a nonzero cyclic code in* $\mathcal{R}_n$. *There exists a polynomial* $g(x) \in \mathcal{C}$ *with the following properties:*

(i) $g(x)$ *is the unique monic polynomial of minimum degree in* $\mathcal{C}$*,*

(ii) $\mathcal{C} = \langle g(x) \rangle$*, and*

(iii) $g(x)|(x^n - 1)$.
Let $k = n - deg\ g(x)$*, and let* $g(x) = \sum_{i=0}^{n-k} g_i x^i$*, where* $g_{n-k} = 1$*. Then:*

(iv) The dimension of $\mathcal{C}$ *is* k *and* $\{g(x), xg(x), \cdots, x^{k-1}g(x)\}$ *is a basis for* $\mathcal{C}$*,*

(v) Every element of $\mathcal{C}$ *is uniquely expressible as a product* $g(x)f(x)$*, where* $f(x) = 0$ *or deg* $f(x) < k$*,*

(vi)

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & & g_0 & \cdots & & & g_{n-k} \end{pmatrix} \leftrightarrow \begin{pmatrix} g(x) & & & \\ & xg(x) & & \\ & & \cdots & \\ & & & x^{k-1}g(x) \end{pmatrix}$$

is a ***generator matrix for*** $\mathcal{C}$*, and*

(vii) If α *is a primitive* n^{th} *root of unity in some extension field of* $\mathbb{F}_q$*, then*

$$g(x) = \prod_s \mathrm{M}_{\alpha^s}(x).$$

where the product is over a subset of representatives of the q*-cyclotomic cosets modulo* n*.*

Proof. See [23]. □

Remark 2.1.1.

(i) $\mathcal{R}_n = \mathbb{F}_q[x]/(x^n - 1)$ is a principal ideal ring.

(ii) Part (vii) requires that $\gcd(n, q) = 1$ because $x^n - 1$ has no repeated roots if and only if $\gcd(n, q) = 1$.

Corollary 2.1.5. *[23] Let $\mathcal{C}$ be nonzero cyclic code in $\mathcal{R}_n$. The following are equivalent:*

(i) $g(x)$ is the monic polynomial of minimum degree in $\mathcal{C}$.

(ii) $\mathcal{C} = \langle g(x) \rangle$, $g(x)$ is monic, and $g(x)|x^n - 1$.

Proof. $(i) \Longrightarrow (ii)$ was shown in the proof of theorem 2.1.4.
Assume (ii). Let $g_1(x)$ be monic polynomial of minimum degree in $\mathcal{C}$. By part (i) and (ii) of theorem 2.1.4 $g(x) = g_1(x)h(x) + r(x)$, $\deg r(x) < \deg g_1(x)$ or $r(x) = 0 \Longrightarrow g_1(x)|g(x)$ in $\mathbb{F}_q[x]$ and $\mathcal{C} = \langle g_1(x) \rangle$. As $g_1(x) \in \mathcal{C} = \langle g(x) \rangle$, $g_1(x) = g(x)a(x) (\text{mod } x^n - 1) \Longrightarrow g_1(x) = g(x)a(x) + (x^n - 1)b(x) \in \mathbb{F}_q[x]$. Since $g(x)|x^n - 1 \Longrightarrow g(x)|g(x)a(x) + (x^n - 1)b(x)$ or $g(x)|g_1(x)$. As both $g_1(x)$ and $g(x)$ are monic and divide one another in $\mathbb{F}_q[x] \Longrightarrow g_1(x) = g(x)$. □

Corollary 2.1.6. *[23] The number of cyclic codes in $\mathcal{R}_n$ equals 2^m, where m is the number of $q-$ cyclotomic cosets modulo n. Moreover, the **dimension of cyclic codes in** $\mathcal{R}_n$ are all possible sums of sizes of the q-cyclotomic cosets modulo n.*

Example 2.1.2. *[23] We showed that, over $\mathbb{F}_2$, $x^9 - 1 = (x+1)(1+x+x^2)(1+x^3+x^6)$, and so there are eight binary cyclic codes $\mathcal{C}_i$ of length 9 with generator polynomials $g_i(x)$ are given in the following table*

i	dim	$g_i(x)$
0	0	$1 + x^9$
1	1	$(1 + x + x^2)(1 + x^3 + x^6) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8$
2	2	$(1 + x)(1 + x^3 + x^6) = 1 + x + x^3 + x^4 + x^6 + x^7$
3	3	$1 + x^3 + x^6$
4	6	$(1 + x)(1 + x + x^2) = 1 + x^3$
5	7	$1 + x + x^2$
6	8	$1 + x$
7	9	1

Corollary 2.1.7. *[23] Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be cyclic codes over $\mathbb{F}_q$ with generator polynomials $g_1(x)$ and $g_2(x)$ respectively. Then $\mathcal{C}_1 \subseteq \mathcal{C}_2$ if and only if $g_2(x)|g_1(x)$.*

Proof. Let $\deg\ g_1(x) = t_1$, $\deg\ g_2(x) = t_2$. If $g_2(x)|g_1(x) \implies \deg\ g_2 \leq \deg\ g_1 \implies -\deg\ g_1 \leq -\deg\ g_2 \implies n - \deg\ g_1 \leq n - \deg\ g_2 \implies \dim\mathcal{C}_1 \leq \dim\mathcal{C}_2$, then $\mathcal{C}_1 \subseteq \mathcal{C}_2$.
$\Longleftarrow$ if $\mathcal{C}_1 \subseteq \mathcal{C}_2 \implies n - \deg\ g_1 \leq n - \deg\ g_2 \implies \deg\ g_2 \leq \deg\ g_1 \implies g_2|g_1$. □

Example 2.1.3. *[23] Consider $\mathcal{R}_3 = \mathbb{F}_2[x]/\langle x^3 - 1\rangle$ and consider the cyclic code $\mathcal{C} = \langle 1 + x\rangle$, then dim $\mathcal{C} = 3 - 1 = 2$ and $\mathcal{C}$ contains the codewords, $0,\ 1 + x,\ x(1 + x) = x + x^2,\ x^2(1 + x) = x^2 + x^3 = x^2 + 1 = (x + 1)(x + 1)$, because $x^3 = 1$. Thus $\mathcal{C} = \{0,\ 1 + x,\ 1 + x^2,\ x + x^2\} = \{000, 110, 101, 011\}$. Also we can verify that*

$$\langle 1 + x^2\rangle = \{f(x)(1 + x^2)|f(x) \in \mathcal{R}_3\} = \mathcal{C}$$

and so $\mathcal{C}$ is generated by the polynomial $1 + x^2$ as well.

Example 2.1.4. *[23] $x^9 - 1$ factors over $\mathbb{F}_2$ into irreducible factors $x^9 - 1 = (x+1)(x^2 + x + 1)(x^6 + x^3 + 1)$.*
Consider $\mathcal{C} = \langle x^6 + x^3 + 1\rangle$, therefore dim $\mathcal{C} = 9 - 6 = 3$ and has generator matrix,

$$G = \begin{pmatrix} x^6 + x^3 + 1 \\ x(x^6 + x^3 + 1) \\ x^2(x^6 + x^3 + 1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Theorem 2.1.8. *[23] The dual code of a cyclic code is cyclic.*

Proof. If $a \cdot b = 0$, then $\pi(a) \cdot \pi(b) = 0$, where π is the cyclic shift. As $a \cdot b = a_0b_0 + a_1b_1 + \cdots + a_nb_n$ then $\pi(a)\cdot\pi(b) = a_nb_n + a_1b_1 + \cdots + a_0b_0 = 0$. Consider the cyclic code $\mathcal{C}$ which is generated by the word v; so $\mathcal{C} = \{v, \pi(v), \pi^(v), \cdots, \pi^{n-1}(v)\}$. If $u \in \mathcal{C}^\perp \implies \pi^i(v) \cdot u = 0$, for $i = 0, \cdots, n - 1$. However this means that $\pi^{i+1}(v).\pi(u) = 0 \implies \pi(u)$ is orthogonal to $\mathcal{C}$, because $\pi^n(v) = v$. Since $u \in \mathcal{C}^\perp \implies \pi(u) \in \mathcal{C}^\perp \implies \mathcal{C}^\perp$ is cyclic. □

Remark 2.1.2. [23] The generator matrix of the dual code of cyclic code $\mathcal{C}$ is the parity check matrix of the original cyclic code.

Theorem 2.1.9. *[23] Let $\mathcal{C}$ be an $[n, k]$ cyclic code with generator polynomial $g(x)$. Let $h(x) = (x^n - 1)/g(x) = \sum_{i=0}^{k} h_ix^i$. Then the generator polynomial of $\mathcal{C}^\perp$ is $g^\perp(x) = \frac{x^k h(x^{-1})}{h(0)}$.*

Furthermore, a generator matrix for $\mathcal{C}^{\perp}$, and hence a parity check matrix for $\mathcal{C}$ is

$$H = \begin{pmatrix} h_k & h_{k-1} & h_{k-2} & \cdots & h_0 & & 0 \\ 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & & h_k & & \cdots & & h_0 \end{pmatrix}$$

Example 2.1.5. *[23] The code $\mathcal{C} = \langle x^6+x^3+1 \rangle$ has generator polynomial $g(x) = x^6+x^3+1$ and has dimension $k = 9-6 = 3$.*

$\therefore h(x) = \frac{x^9-1}{g(x)} = (1+x)(1+x+x^2) = 1+x^3$.

The generator polynomial of $\mathcal{C}^{\perp}$ is $g^{\perp}(x) = \frac{x^k h(x^{-1})}{h(0)} = \frac{x^3(1+x^{-3})}{1} = x^3+1$,

$\therefore$ *The generator matrix for $\mathcal{C}^{\perp}$ is*

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

Remark 2.1.3. [23]

(i) The polynomial $h(x)$ is called the **check polynomial for a cyclic code** $\mathcal{C}$. It is not the generator polynomial for $\mathcal{C}^{\perp}$.

(ii) The polynomial $g^{\perp}(x) = \frac{x^k h(x^{-1})}{h(0)}$ is called the **reverse polynomial of the check polynomial** $h(x)$ and $g^{\perp}(x)$ is the **generator polynomial for** $\mathcal{C}^{\perp}$.
If $h(x) = h_0 + h_1 x + \cdots + h_k x^k$, then the **generator polynomial of** $\mathcal{C}^{\perp}$ is
$g^{\perp}(x) = \frac{x^k h(x^{-1})}{h(0)} = h^{-1}(0)(x^k)(h_0 + h_1 x^{-1} + \cdots + h_k x^{-k})$
$= h^{-1}(0)(h_0 x^k + h_1 x^{k-1} + \cdots + h_k) = h^{-1}(0)(h_k + h_{k-1} + \cdots + h_0 x^k)$

2.2 Encoding and decoding cyclic codes

In this section, we study three ways of encoding cyclic codes, one of there methods is called non systematic, and the other two methods are called systematic.

★ Encoding cyclic codes by non systematic method

Let $\mathcal{C}$ be a cyclic code of length n over $\mathbb{F}_q$ with generator polynomial $g(x)$ of degree $n-k$, so $\mathcal{C}$ has dimension k, Let G be the generator matrix as describe in Previous section

$$G = \begin{pmatrix} g(x) \\ xg(x) \\ \cdots \\ x^{k-2}g(x) \end{pmatrix} = \begin{pmatrix} g_0 & \cdots & g_{n-k} & & 0 \\ 0 & g_0 & \cdots & g_{n-k} & \\ & \cdots & & \cdots & \\ 0 & & g_0 & \cdots & g_{n-k} \end{pmatrix}.$$

To encode the message $m \in \mathbb{F}_q^k$ as the codeword $\mathbf{c} = mG$, let $m(x) = a_0 + a_1x + \cdots + a_{k-1}x^k \in \mathbb{F}_q[x]$. Then to encode $m(x)$ as a codeword $\mathbf{c}(x)$ by forming the product $\mathbf{c}(x) = m(x)g(x)$. Which called first encoding procedure, See [23].

Example 2.2.1. *[23] Let $\mathcal{C}$ be a cyclic code of length 15 with generator polynomial $g(x) = (1+x+x^4)(1+x+x^2+x^3+x^4)$. Encode the message $m(x) = 1+x^2+x^5$ using the first encoding procedure.*

Solution: $g(x) = (1+x+x^4)(1+x+x^2+x^3+x^4) = 1+x^4+x^6+x^7+x^8$. *Then* $\mathbf{c}(x) = m(x)g(x) = (1+x^2+x^5)(1+x^4+x^6+x^7+x^8) = 1+x^2+x^4+x^5+x^7+x^{10}+x^{11}+x^{12}+x^{13}$ *as a vector in* $\mathbb{F}_2^{15} = (101011010011110)$.

★ Encoding cyclic codes by systematic method

(i) The polynomial $m(x)$ associated to the message is of degree at most $k-1$, or it is the zero polynomial. The polynomial $x^{n-k}m(x)$ has degree at most $n-1$ and has its first $n-k$ coefficients equal to 0. Thus the message is contained in the coefficients of $x^{n-k}, x^{n-k+1}, \cdots, x^{n-1}$. By the Division Algorithm, $x^{n-k}m(x) = g(x)a(x) + r(x)$,

where deg $r(x) < n-k$ or $r(x) = 0$.

Let $\mathbf{c}(x) = x^{n-k}m(x) - r(x)$, as $\mathbf{c}(x)$ is a multiple of $g(x)$, $\mathbf{c}(x) \in \mathcal{C}$. Also $\mathbf{c}(x)$ differes from $x^{n-k}m(x)$ in the coefficients of $1, x, \cdots, x^{n-k-1}$ as deg $r(x) < n-k$. So $\mathbf{c}(x)$ contains the message m in the coefficients of the terms of degree at least $n-k$.

Example 2.2.2. *[23] Let $\mathcal{C}$ be as in example 2.2.1 Encode the message $m(x) = 1 + x^2 + x^5$ using the second encoding procedure.(The systematic encoding).*

Solution:*Since from previous example $g(x) = 1 + x^4 + x^6 + x^7 + x^8$, $n = 15$, $k = n -$ deg $g(x) = 15 - 8 = 7$*

$\therefore x^{n-k} = x^{15-7} = x^8 \Longrightarrow x^{n-k}m(x) = x^8(1 + x^2 + x^5) = x^8 + x^{10} + x^{13}$.

By dividing $x^8 + x^{10} + x^{13}$ by $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ we get

$x^{13} + x^{10} + x^8 = (x^5 + x^4 + x + 1)g(x) + (x^6 + x + 1)$ *where* $a(x) = x^5 + x^4 + x + 1$ *and* $r(x) = x^6 + x + 1$.

$\mathbf{c}(x) = x^{13} + x^{10} + x^8 + x^6 + x + 1$. *The message is contained in the coefficients x^{n-k}, $x^{n-k+1}, \cdots, x^{n-1}$ (systematic).*

(ii) Let $\mathcal{C} = \langle g(x) \rangle$ be a cyclic code. Let $g^{\perp}(x)$ be the generator polynomial of $\mathcal{C}^{\perp}$ and $\mathcal{C}$ is an $[n, k]$-code. If $\mathbf{c} = (c_0 c_1 \cdots c_{n-1}) \in \mathcal{C}$ once $c_0 c_1 \cdots c_{k-1}$ are known, then the remaining components $c_k, \cdots, c_{n-1}$ are determined form $H\mathbf{c}^{\perp} = 0$, where H is the parity check matrix. We can scale the rows of H so that its rows are shifts of the monic polynomial $g^{\perp}(x) = h'_0 + h'_1 x + \cdots + h'_{k-1}x^{k-1} + x^k$. To encode $\mathcal{C}$, we chose k information bits $c_0 c_1 \cdots c_{k-1}$, then $c_i = -\sum_{j=0}^{k-1} h'_j c_{i-k+j}$, where the computation c_i is performed in the order $i = k, k+1, \cdots, n-1$.

Example 2.2.3. *[23] Encode the message $m(x) = 1 + x^2 + x^5$ using the third encoding procedure.*

Solution: $h(x) = \frac{x^{15}-1}{g(x)}$, *where* $g(x) = 1 + x^4 + x^6 + x^7 + x^8$. *Then* $h(x) = x^7 + x^6 + x^4 + 1$.

$n = 15$, *deg* $g(x) = 8 \Longrightarrow k = 7$.

$g^{\perp}(x) = x^k h(x^{-1})/h(0) = x^7(x^{-7} + x^{-6} + x^{-4} + 1)/1 = 1 + x + x^3 + x^7.$

$h'_0 = 1, h'_1 = 1, h'_2 = 0, h'_3 = 1, h'_4 = 0, h'_5 = 0, h'_6 = 0, h'_7 = 1.$

$$H = \begin{pmatrix} 1 + x + x^3 + x^7 \\ xg^{\perp}(x) = x(1 + x + x^3 + x^7) \\ \vdots \\ x^7(1 + x + x^3 + x^7) \end{pmatrix} = \begin{pmatrix} 1 + x + x^3 + x^7 \\ x + x^2 + x^4 + x^8 \\ x^2 + x^3 + x^5 + x^9 \\ x^3 + x^4 + x^6 + x^{10} \\ x^4 + x^5 + x^7 + x^{11} \\ x^5 + x^6 + x^8 + x^{12} \\ x^6 + x^7 + x^9 + x^{13} \\ x^7 + x^8 + x^{10} + x^{14} \end{pmatrix}$$

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

Let the information bits are $(c_0 c_1 \cdots c_6)$, *since* $m(x) = 1 + x^2 + x^5 \longleftrightarrow (1010010)$.

Then

$c_7 = -\sum_{j=0}^{6} h'_j c_{7-7+j} = +(h'_0 c_0 + h'_1 c_1 + h'_2 c_2 + \cdots + h'_6 c_6) = (c_0 + c_1 + 0 + c_3 + 0 + 0 + 0) = (c_0 + c_1 + c_3)$

$c_8 = \sum_{j=0}^{6} h'_j c_{8-7+j} = +(h'_0 c_1 + h'_1 c_2 + h'_2 c_3 + \cdots + h'_6 c_7) = (c_1 + c_2 + c_4)$

similarly we do for all other bits.

2.3 Computing the generating idempotent e(x) for a cyclic code

In section 1.4 we define generating idempotent $e(x)$ for a cyclic code and give some properties for $e(x)$, here want to get a method to find generating idempotent $e(x)$ by solve $1 = a(x)g(x)+b(x)h(x)$ for $a(x)$ using the Euclidean Algorithm, where $h(x) = \frac{x^n-1}{g(x)}$. Then reducing $a(x)g(x)$ modulo $x^n - 1$ produces $e(x)$.

Also we can compute $g(x)$ by Theorem 1.4.3 if $e(x)$ known.

Example 2.3.1. *As the same way in Example 2.1.2 the following table gives all the cyclic codes $\mathcal{C}_i$ of length 7 over $\mathbb{F}_2$ together with their generator polynomials $g_i(x)$ and their generating idempotents $e_i(x)$.*

i	dim	$g_i(x)$	$e_i(x)$
0	0	$1 + x^7$	0
1	1	$1 + x + x^2 + x^3 + x^4 + x^5 + x^6$	$1 + x + x^2 + x^3 + x^4 + x^5 + x^6$
2	3	$1 + x^2 + x^3 + x^4$	$1 + x^3 + x^5 + x^6$
3	3	$1 + x + x^2 + x^4$	$1 + x + x^2 + x^4$
4	4	$1 + x + x^3$	$x + x^2 + x^4$
5	4	$1 + x^2 + x^3$	$x^3 + x^5 + x^6$
6	6	$1 + x$	$x + x^2 + x^3 + x^4 + x^5 + x^6$
7	7	1	1

⋆ ***To find*** $e_1(x)$ *if* $g(x) = 1+x+x^2+x^3+x^4+x^5+x^6$ *we first find* $h(x) = \frac{x^7+1}{g(x)} = x+1$. *Then we form the equation* $1 = a(x)g(x) + b(x)h(x)$ *by using Euclidean Algorithm, we divide* $g(x)$ *by* $h(x)$ *we get*

$g(x) = (x + 1)(x^5 + x^3 + x) + 1$, *then* $1 = 1 \cdot g(x) + (x + 1)(x^5 + x^3 + x)$, *where* $b(x) = (x^5 + x^3 + x)$ *and* $a(x) = 1$

$\therefore e(x) = 1 \cdot g(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6$.

⋆ ***To find*** $e_2(x)$ *if* $g(x) = 1 + x^2 + x^3 + x^4$ *we first find* $h(x) = \frac{x^7+1}{g(x)} = x^3 + x^2 + 1$.

Divide $g(x)$ *by* $h(x)$ *we get*

$1 + x^2 + x^3 + x^4 = (x^3 + x^2 + 1) \cdot x + (x^2 + x + 1)$ *using Euclidean algorithm*

$(x^3 + x^2 + 1) = (x^2 + x + 1) \cdot x + (x + 1)$

$(x^2 + x + 1) = (x + 1) \cdot x + 1$

Now want to get $a(x)$ *and* $b(x)$ *as follows:*

$1 = (x^2 + x + 1) + (x + 1) \cdot x$

$= (x^2 + x + 1) + x \cdot [h(x) + x \cdot (x^2 + x + 1)]$

$= (x^2 + x + 1) + x \cdot h(x) + x \cdot x \cdot (x^2 + x + 1)$

$= (x^2 + x + 1) + x \cdot h(x) + x^2[g(x) + x \cdot h(x)]$

$= (x^2 + x + 1) + x \cdot h(x) + x^2 \cdot g(x) + x^3 \cdot h(x)$

$= g(x) + x \cdot h(x) + x \cdot h(x) + x^2 \cdot g(x) + x^3 \cdot h(x)$

$= (x^2 + 1) \cdot g(x) + x^3 \cdot h(x)$

So $e_2(x) = (x^2 + 1) \cdot g(x)$

$= (x^2 + 1) \cdot (1 + x^2 + x^3 + x^4)$

$= x^6 + x^5 + x^3 + 1.$

$\star$ ***To find*** $e_3(x)$ *if* $g(x) = 1 + x + x^2 + x^4$ *we first find* $h(x) = \frac{x^7+1}{g(x)} = x^3 + x + 1.$

Divide $g(x)$ *by* $h(x)$ *we get*

$1 + x + x^2 + x^4 = x \cdot x^3 + x + 1 + 1$

$1 = 1 \cdot g(x) + x \cdot h(x)$

So $e_3(x) = 1 \cdot g(x) = 1 + x + x^2 + x^4$.

Continue in the same way to get the generating idempotents for each $g_i(x)$

Theorem 2.3.1. *[23] Let* $\mathcal{C}$ *be an* $[n, k]$ *cyclic code with generating idempotent* $e(x) = \sum_{i=0}^{n-1} e_i x^i$. *Then the* $k \times n$ *matrix*

$$\begin{pmatrix} e_0 & e_1 & e_2 & \cdots & e_{n-2} & e_{n-1} \\ e_{n-1} & e_0 & e_1 & \cdots & e_{n-3} & e_{n-2} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ e_{n-k+1} & e_{n-k+2} & \cdots & \cdots & e_{n-k-1} & e_{n-k} \end{pmatrix}$$

is a generator matrix for $\mathcal{C}$.

Definition 2.3.1. [23] If $\mathcal{C}_1$ and $\mathcal{C}_2$ are codes of length n over $\mathbb{F}_q$, then $\mathcal{C}_1 + \mathcal{C}_2 = \{\mathbf{c}_1 + \mathbf{c}_2 | \mathbf{c}_1 \in \mathcal{C}_1 \text{ and } \mathbf{c}_2 \in \mathcal{C}_2\}$ is the sum of $\mathcal{C}_1$ and $\mathcal{C}_2$.

Notation 2.3.2. *Both the sum of and intersection of two cyclic codes are cyclic.*

2.4 Meggitt decoding of cyclic codes

In this section,we give a new decoding method of cyclic code,There are several versions of Meggitt decoding; we will present one of them. We use a useful program called ***Maple 13*** *to factorize the polynomial $x^n - 1$ mod 2, find the remainder when dividing polynomial by another polynomial and we will look to the applications of* ***Maple 13*** *more carefully in the last two chapters.*

Let $\mathcal{C}$ be an $[n, k, d]$ cyclic code over $\mathbb{F}_q$ with generator polynomial $g(x)$ of degree $n-k$; $\mathcal{C}$ will correct $t = \lfloor \frac{(d-1)}{2} \rfloor$ errors. Suppose that $c(x) \in \mathcal{C}$ is transmitted and $y(x) = c(x)+e(x)$ is received, where $e(x) = e_0 + e_1 x + \cdots + e_{n-1}x^{n-1}$ is the error vector with $wt(e(x)) \leq t$. The Meggitt decoder stores syndromes of error patterns with coordinate $n-1$ in error. The first version of Meggitt decoding algorithm described by shifting $y(x)$ at most n times, the decoder finds the error vector $e(x)$ from the list and corrects the errors.

In the second version, by shifting $y(x)$ until an error appears in coordinate $n-1$, the decoder finds the error in that coordinate, correct only that error, and then corrects errors in coordinates $n-2, n-3, \cdots, 1, 0$ in that order by further shifting.

For any vector $v(x) \in \mathbb{F}_q[x]$, let $\mathcal{R}_{g(x)}(v(x)$ be the unique remainder when $v(x)$ is divided by $g(x)$ according to the Division Algorithm, that is $\mathcal{R}_{g(x)})(v(x)) = r(x)$ **where** $v(x) = g(x)f(x) + r(x)$**, where** $r(x) = 0$ **or** *deg* $r(x) < n-k$**. The function** $\mathcal{R}_{g(x)}(v(x))$ satisfies the following properties,

Theorem 2.4.1. *[23] With the preceding notation the following hold:*

(i) $\mathcal{R}_{g(x)}(av(x) + bv'(x)) = a\mathcal{R}_{g(x)}(v(x)) + b\mathcal{R}_{g(x)}(v'(x))$ *for all* $v(x), v'(x) \in \mathbb{F}_q[x]$ *and all* $a, b \in \mathbb{F}_q$.

(ii) $\mathcal{R}_{g(x)}(v(x) + a(x)(x^n - 1)) = \mathcal{R}_{g(x)}(v(x))$

(iii) $\mathcal{R}_{g(x)}(v(x)) = 0$ *if and only if* $v(x)$ *mod* $(x^n - 1) \in \mathcal{C}$.

(iv) If $c(x) \in \mathcal{C}$, *then* $\mathcal{R}_{g(x)}(c(x) + e(x)) = \mathcal{R}_{g(x)}(e(x))$

(v) If $\mathcal{R}_{g(x)}(e(x)) = \mathcal{R}_{g(x)}(e'(x))$, *where* $e(x)$ *and* $e'(x)$ *each have weight at most* t, *then* $e(x) = e'(x)$

(vi) $\mathcal{R}_{g(x)}(v(x)) = v(x)$ *if deg* $v(x) < n - k$.

Proof. (i) Let $r(x) = \mathcal{R}_{g(x)}(v(x))$, $r'(x) = \mathcal{R}_{g(x)}(v'(x))$ where $v(x) = g(x)f(x) + r(x)$, $v'(x) = g(x)f'(x) + r'(x)$ with $r(x) = 0$ or $deg\ r(x) < n - k$, and $r'(x) = 0$ or $deg\ r'(x) < n - k$.

$r(x) = v(x) - g(x)f(x)$, $r'(x) = v'(x) - g(x)f'(x)$..

$ar(x) = av(x) - ag(x)f(x)$, $br'(x) = bv'(x) - bg(x)f'(x)$

$ar(x) + br'(x) = av(x) - ag(x)f(x) + bv'(x) - bg(x)f'(x)$

$av(x) + bv'(x) = ag(x)f(x) + bg(x)f'(x) + ar(x) + br'(x) = g(x)(af(x) + bf'(x)) + ar(x) + br'(x)$, where $deg\ ar(x) + br'(x) < n - k$ or $ar(x) + br'(x) = 0 \implies \mathcal{R}_{g(x)}(av(x)+bv'(x)) = ar(x)+br'(x) = a\mathcal{R}_{g(x)}(v(x))+b\mathcal{R}_{g(x)}(v'(x))$ for all $v(x)$, $r'(x) \in \mathbb{F}_q[x]$ and $a, b \in \mathbb{F}_q$.

(ii) Let $v(x) = g(x)f(x) + r(x)$ where $r(x) = 0$ or $deg\ r(x) < n - k$. $\mathcal{R}_{g(x)}(v(x)) = r(x) = v(x) - g(x)f(x)$ since $g(x)$ is a generator polynomial, then $g(x)|x^n - 1 \implies x^n - 1 = k(x)g(x) \implies a(x)(x^n - 1) = a(x)k(x)g(x) \implies v(x) + a(x)(x^n - 1) = g(x)f(x) + r(x) + a(x)k(x)g(x) = g(x)[f(x) + a(x)k(x)] + r(x)$.

$r(x) = \mathcal{R}_{g(x)}(v(x)) = \mathcal{R}_{g(x)}(v(x) + a(x)(x^n - 1))$.

(iii) If $r(x) = \mathcal{R}_{g(x)}(v(x)) = 0 \implies v(x) = g(x)f(x) \implies v(x)+a(x)(x^n-1) \equiv g(x)f(x)+a(x)k(x)g(x) = g(x)(f(x) + a(x)k(x)) \in \mathcal{C} \implies v(x)\ mod\ (x^n - 1) \in \mathcal{C}$.

Conversely, if $v(x)(mod\ x^n-1) \in \mathcal{C} \implies v(x)+a(x)(x^n-1) \in \mathcal{C} \implies \mathcal{R}_{g(x)}(v(x)) = 0$.

(iv) If $c(x) \in \mathcal{C}$, then $\mathcal{R}_{g(x)}(c(x)) = 0$ and $\implies \mathcal{R}_{g(x)}(c(x))+\mathcal{R}_{g(x)}(e(x)) = 0+\mathcal{R}_{g(x)}(e(x))$

(v) If $\mathcal{R}_{g(x)}(e(x)) = \mathcal{R}_{g(x)}(e'(x))$ then there exists $c(x) \in \mathcal{C}$ such that $\mathcal{R}_{g(x)}(c(x) + e(x)) = \mathcal{R}_{g(x)}(e(x))$ and $\mathcal{R}_{g(x)}(c(x) + e'(x)) = \mathcal{R}_{g(x)}(e'(x)) \implies \mathcal{R}_{g(x)}(c(x) + e'(x)) -$

$\mathcal{R}_{g(x)}(e'(x)) = \mathcal{R}_{g(x)}(e(x) - e'(x)) = 0 \Longrightarrow e(x) - e'(x)\ mod\ (x^n - 1) \in \mathcal{C}$ where $deg\ (e(x) - e'(x) < n - k$ but $g(x)$ is a unique monic polynomial of minimal degree $n - k \Longrightarrow e(x) - e'(x) = 0 \Longrightarrow e(x) - e'(x)$.

(vi) $\mathcal{R}_{g(x)}(v(x)) = v(x)$ if $deg\ v(x) < n - k$.

If $deg\ v(x) < n - k \Longrightarrow v(x) = 0 \cdot g(x) + v(x) \Longrightarrow \mathcal{R}_{g(x)}(v(x)) = v(x)$.

□

Theorem 2.4.2. *[23] Let $g(x)$ be a monic divisor of $x^n - 1$ of degree $n-k$. If $\mathcal{R}_{g(x)}(v(x) = S(x)$, then $\mathcal{R}_{g(x)}(xv(x))mod\ (x^n-1)) = \mathcal{R}_{g(x)}(xS(x)) = xS(x) - g(x)S_{n-k-1}$, where S_{n-k-1} is the coefficient of x^{n-k-1} in $S(x)$.*

Proof. By definition $v(x) = g(x)f(x) + S(x)$ where $S(x) = \sum_{i=0}^{n-k-1} S_i x^i$. So $xv(x) = xg(x)f(x) + xS(x) = xg(x) \cdot f(x) + g(x)f_1(x) + S'(x)$, where $S'(x) = \mathcal{R}_{g(x)}(xS(x))$. Also $xv(x)\ mod\ (x^n - 1) = xv(x) - (x^n - 1)v_{n-1}$. (Because $xv(x) = v_0x + \cdots + v_{n-1}x^n \Longrightarrow xv(x)\ (mod\ x^n - 1) = xv(x) - (x^n - 1)$. Thus $xv(x)\ mod\ (x^n - 1) = xg(x)f(x) + g(x)f(x)f_1(x) + S'(x) - (x^n - 1)v_{n-1} = (xf(x) + f_1(x) - h(x)v_{n-1})g(x) + S'(x)$, where $g(x)h(x) = x^n - 1$. Therefore $\mathcal{R}_{g(x)}(xv(x))\ mod\ (x^n - 1) = S'(x) = \mathcal{R}_{g(x)}(xS(x))$, because $deg\ S'(x) < n - k = deg\ g(x)$. As $g(x)$ is monic of degree $n - k$ and $xS(x) \sum_{i=0}^{n-k-1} S_i x^{i+1}$, the remainder when $xS(x)$ is divided by $g(x)$ is $xS(x) - g(x)S_{n-k-1}$, (because $xS(x) = S_0x + S_1x^2 + \cdots + S_{n-k-1}x^{n-k}) = g(x)S_{n-k-1} + \mathcal{R}_{g(x)}(xS(x)) \Longrightarrow \mathcal{R}_{g(x)}(xS(x)) = xS(x) - g(x)S_{n-k-1}$. □

Definition 2.4.1. [23] The weight of a polynomial is the number of nonzero coefficients.

Definition 2.4.2. [23] The syndrome $S(v(x))$ of $v(x)$ is defined by $S(v(x)) = \mathcal{R}_{g(x)}(x^{n-k}v(x))$.

Remark 2.4.1. If $v(x) \in \mathcal{R}_n$, then $S(v(x)) = 0$ if and only if $v(x) \in \mathcal{C}$.

★ The first version of Meggitt decoding algorithm

Step I: We find all the syndrome polynomials $S(e(x))$ of error patterns $e(x) = \sum_{i=0}^{n-1} e_i x^i$ such that $wt(e(x)) \leq t$ and $e_{n-1} \neq 0$.

Example 2.4.1. *Let $\mathcal{C}$ be the $[15, 7, 5]$ binary cyclic code . Let α be a $15th$ root of unity in $\mathbb{F}_{16}$. We can factor $x^{15} - 1$ by* ***Maple 13*** *as follows:*

$$Factors(x^{15} - 1) \ \ mod \ \ 2$$

To get that

$$[1, [[x^4 + x + 1, 1], [x + 1, 1], [x^2 + x + 1, 1], [x^4 + x^3 + x^2 + x + 1, 1], [x^4 + x^3 + 1, 1]]]$$

i.e

$$x^{15} - 1 = (x^4 + x + 1)(x + 1)(x^2 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^4 + x^3 + 1).$$

Then $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ is the generator polynomial of $\mathcal{C}$ and the syndrome of $e(x)$ is $S(e(x)) = \mathcal{R}_{g(x)}(x^8 e(x))$. Step I produces the following syndrome polynomials:

$e(x)$	$S(e(x))$
x^{14}	x^7
$x^{13} + x^{14}$	$x^6 + x^7$
$x^{12} + x^{14}$	$x^5 + x^7$
$x^{11} + x^{14}$	$x^4 + x^7$
$x^{10} + x^{14}$	$x^3 + x^7$
$x^9 + x^{14}$	$x^2 + x^7$
$x^8 + x^{14}$	$x + x^7$
$x^7 + x^{14}$	$1 + x^7$
$x^6 + x^{14}$	$x^3 + x^5 + x^6$
$x^5 + x^{14}$	$x^2 + x^4 + x^5 + x^6 + x^7$
$x^4 + x^{14}$	$x + x^3 + x^4 + x^5 + x^7$
$x^3 + x^{14}$	$1 + x^2 + x^3 + x^4 + x^7$
$x^2 + x^{14}$	$x + x^2 + x^5 + x^6$
$x + x^{14}$	$1 + x + x^4 + x^5 + x^6 + x^7$
$1 + x^{14}$	$1 + x^4 + x^6$

for example to compute $S(x^{12}+x^{14}) = \mathcal{R}_{g(x)}(x^8(x^{12}+x^{14})) = \mathcal{R}_{g(x)}(x^{20}+x^{22}) = \mathcal{R}_{g(x)}(x^5+x^7) = x^5+x^7$ because $deg\ x^7+x^7 < 8 = deg\ g(x)$ also $S(1+x^{14}) = \mathcal{R}_{g(x)}(x^8(1+x^{14})) = \mathcal{R}_{g(x)}(x^8+x^7) = \mathcal{R}_{g(x)}(x^8) + \mathcal{R}_{g(x)}(x^7) = (1+x^4+x^6+x^7)+x^7 = 1+x^4+x^6$. For $\mathcal{R}_{g(x)}(x^9) = \mathcal{R}_{g(x)}(xx^8) \Rightarrow \mathcal{R}_{g(x)}(x^9) = \mathcal{R}_{g(x)}(x(1+x^4+x^6+x^7)) = \mathcal{R}_{g(x)}(x+x^5+x^7+x^8) = \mathcal{R}_{g(x)}(x+x^5+x^7+1+x^4+x^6+x^7) = \mathcal{R}_{g(x)}(1+x+x^4+x^5+x^6) = 1+x+x^4+x^5+x^6$. For $S(x+x^{14}) = \mathcal{R}_{g(x)}(x^8(x+x^{14}) = \mathcal{R}_{g(x)}(x^9+x^7) = \mathcal{R}_{g(x)}(x^9) + \mathcal{R}_{g(x)}(x^7) = 1+x+x^4+x^5+x^6+x^7$. The others follow similarly.

Step II: Suppose that $y(x)$ is the received vector, compute the syndrome $S(y(x)) = \mathcal{R}_{g(x)}(x^{n-k}y(x))$, since $y(x) = c(x)+e(x)$, with $c(x) \in \mathcal{C} \Longrightarrow S(y(x)) = S(c(x)) + S(e(x)) = 0 + S(e(x))$.

Step III: If $S(y(x))$ is in the list computed in step I, then you know the error polynomial $e(x)$ and this can be subtracted from $y(x)$ to obtain the codeword $c(x)$. If $S(y(x))$ is not in the list go on step (iv).

Step iv: Compute the syndrome polynomial of $xy(x)$, $x^2y(x), \cdots$ in succession until the syndrome polynomial is in the list from step I. If $S(x^iy(x))$ is in the list and is associated with the error polynomial $e'(x)$, then the received vector is decoded as $y(x) - x^{n-i}e'(x)$. The computation in step (iv) is most easily carried out as

$\mathcal{R}_{g(x)}(x^{n-k}y(x)) = S(y(x)) = \sum_{i=0}^{n-k-1} S_ix^i$,

$S(xy(x)) = \mathcal{R}_{g(x)}(x^{n-k}xy(x)) = \mathcal{R}_{g(x)}(x(x^{n-k}y(x))) = \mathcal{R}_{g(x)}(xS(y(x))) = xS(y(x)) - S_{n-k-1}g(x)$

Example 2.4.2. *Let $\mathcal{C}$ be the [15, 7, 5] binary cyclic code with generator polynomil $g(x) = 1+x^4+x^6+x^7+x^8$. Let $y(x) = 1+x+x^6+x^9+x^{11}+x^{12}+x^{13}$ is received. Carry out steps (I-IV) in the first version of Miggitt decoding algorithm to decode $y(x)$.*

Solution: *In previous example we found all the syndrome polynomials $S(e(x))$ of error patterns $e(x)$. Now want to compute $S(y(x)) = \mathcal{R}_{g(x)}(x^8y(x))$*

$= \mathcal{R}_{g(x)}(x^8(1+x+x^6+x^9+x^{11}+x^{12}+x^{13}))$

$= \mathcal{R}_{g(x)}(x^8+x^9+x^{14}+x^{17}+x^{19}+x^{20}+x^{21}))$

Divide $x^8 + x^9 + x^{14} + x^{17} + x^{19} + x^{20} + x^{21}$ *by* $g(x)$

in ***Maple 13*** *as follows:*

$$rem(x^{21} + x^{20} + x^{19} + x^{17} + x^{14} + x^9 + x^8, x^8 + x^7 + x^6 + x^4 + 1, x) \mod 2$$

To get the result directly $x^5 + x^4 + x^3 + x^2 + x$

i.e $S(y(x)) = x^5 + x^4 + x^3 + x^2 + x$

$S(xy(x)) = xS(y(x)) - S_{n-k-1}g(x)$

$= x \cdot (x^5 + x^4 + x^3 + x^2 + x) - S_{15-7-1} \cdot (1 + x^4 + x^6 + x^7 + x^8)$

$= x^6 + x^5 + x^4 + x^3 + x^2) + S_7 \cdot (1 + x^4 + x^6 + x^7 + x^8)$

$= x^6 + x^5 + x^4 + x^3 + x^2$. *Which is not in the list.*

$S(x^2 \cdot y(x)) = xS(x \cdot y(x)) - S_7 \cdot g(x)$

$= x \cdot (x^6 + x^5 + x^4 + x^3 + x^2) - 0$

$= x^7 + x^6 + x^5 + x^4 + x^3)$. *Which is not in the list.*

$S(x^3 \cdot y(x)) = xS(x^2 \cdot y(x)) - S_7 \cdot g(x)$

$= x^8 + x^7 + x^6 + x^5 + x^4 - 1 \cdot (1 + x^4 + x^6 + x^7 + x^8)$

$= x^5 + 1$. *Which is not in the list.*

$S(x^4 \cdot y(x)) = xS(x^3 \cdot y(x)) - S_7 \cdot g(x)$

$= x^6 + x - 0$

$= x^6 + x$. *Which is not in the list.*

$S(x^5 \cdot y(x)) = xS(x^4 \cdot y(x)) - S_7 \cdot g(x)$

$= x^7 + x^2 - 0$

$= x^7 + x^2$. *Which is in the list and corresponds to the error* $x^9 + x^{19}$

$\therefore y(x)$ *is decode as:*

$y(x) - x^{n-i}e'(x) = y(x) - (x^{15-5}) \cdot (x^9 + x^{14})$

$= y(x) - (x^{10}) \cdot (x^9 + x^{14})$

$= y(x) - (x^{19} + x^{24}) \mod 15$

$= y(x) - (x^4 + x^9)$

$= 1 + x + x^4 + x^6 + x^{11} + x^{12} + x^{13}$

Chapter 3

Decoding binary quadratic residue codes of prime length

A series of different algebraic decoding algorithm for the quadratic residue (QR) codes which was introduced by Prange [13] in 1958, have been proposed in [14], [15], [16] and [17]. In this chapter we will use ***Maple 13*** *more carefully to get the important result to decode the binary quadratic residue code of odd prime like* $41, 47, 73$. *This chapter is divided into four sections. In section one, we set some definitions and theorems which will be applied throughout the thesis. In section two, we study a new algebraic decoding method for the binary* $(41, 21, 9)$ *QR code. In section three and four we study a decoding method for the binary* $(47, 24, 11)$ *and* $(73, 37, 13)$ *quadratic residue codes.*

3.1 Background of the binary quadratic residue code

A binary QR code of length n is an (n, (n+1)/2) cyclic code over $GF(2^m)$ with generator polynomial $g(x)$ where m is some integer. The length of this code is a prime number of the form $n = 8l + 1$ where l is some integer. The generator polynomial $g(x)$ is defined by

$$g(x) = \prod_{i \in Q_n} (x - \beta^i). \qquad (1)$$

where β is a primitive nth root of unity in the finite field $GF(2^m)$ with m being the smallest positive integer such that $n|2^m - 1$ and Q_n is the collection of **all nonzero quadratic residues modulo n** given by

$$Q_n = \{i | i \equiv j^2 \ mod \ n \ for \ 1 \leq j \leq n-1\}. \qquad (2)$$

Let a codeword $c(x) = m(x)g(x)$ be transmitted through a noisy channel to obtain a received codeword of the form $r(x) = c(x) + e(x) = \sum_{i=1}^{n-1} c_i x^i + \sum_{i=1}^{n-1} e_i x^i$, where e(x) is the polynomial of the received error pattern vector. The **syndromes** S_i are defined by

$$S_i = r(\beta^i) = e(\beta^i), \quad for \ i \in Q_n. \qquad (3)$$

Otherwise, the syndromes are called the unknown syndromes and are not obtained directly.

The aim of decoding is to find the ν unknown error locations from the known syndromes S_i for $i \in Q$. For this purpose the **error-locator polynomial** $L(z)$ for every correctable error pattern is defined by

$$L(z) = \prod_{i=1}^{\nu}(z - Z_i) = z^\nu + \sum_{j=1}^{\nu} \sigma_j z^{\nu-j}. \qquad (4)$$

Here, Z_j for $1 \leq j \leq \nu$ are the locations of the ν errors, i.e. $Z_j = \alpha^{r_j}$, where r_j locates the position of the error to be corrected and $\nu < t$.

$\star$To decode a QR code it suffices to find the error-locator polynomial $L(z)$. It is well known that S_i and σ_j are related by the following **Newton identities** [5]

$$S_i + \sum_{j=1}^{i-1} \sigma_j S_{i-j} + \sigma_j = 0 \quad (1 \leq i \leq \nu, i = odd)$$

$$S_i + \sum_{j=1}^{i-1} \sigma_j S_{i-j} = 0 \quad (1 \leq i \leq \nu, i = even)$$

$$S_i + \sum_{j=1}^{\nu} \sigma_j S_{i-j} = 0 \quad (i \geq \nu) \qquad (5)$$

Next, in order to simplify the decoding process, another set of identities related to S_{-i} and σ_j given in the following

$$S_{-i} + \sum_{j=1}^{i-1} \frac{\sigma_{\nu-j}}{\sigma_\nu} S_{-(i-j)} + \frac{\sigma_{\nu-i}}{\sigma_\nu} = 0 \quad (1 \le i \le \nu, i = odd)$$

$$S_{-i} + \sum_{j=1}^{i-1} \frac{\sigma_{\nu-j}}{\sigma_\nu} S_{-(i-j)} = 0 \quad (1 \le i \le \nu, i = even)$$

$$S_{-i} + \sum_{j=1}^{\nu} \frac{\sigma_{\nu-j}}{\sigma_\nu} S_{-(i-j)} = 0 \quad (i \ge \nu) \qquad (6)$$

For a t-error-correcting QR code that if there are $\nu \le t$ errors in the received codeword, one must have $\sigma_{\nu+1} = \sigma_{\nu+2} = \sigma_t = 0$, and $S_0 \equiv \nu \bmod 2$.

If there is a sufficient number of consecutive known syndromes for a given number of errors, one can directly solve from the Newtons identities for the σ_j, $1 \le j \le \nu$. However, if there are not enough consecutive syndromes, one first tries to find the unknown syndromes and then to find $L(z)$ from the Newton identities. In either case, once $L(z)$ is found, the error pattern is found by a search of the roots of $L(z)$ over the set of all the n^{th} roots of unity.

If not, Assume that ν errors occur in the received vector. Let $I = \{i_1, i_2, \ldots, i_{\nu+1}\}$ and $J = \{j_1, j_2, \ldots, j_{\nu+1}\}$ denote two subsets of $\{1, 2, \ldots, n-1\}$. The matrices $X(I)$ and $X(J)$ are defined to have the following forms:

$$X(I) = \begin{pmatrix} Z_1^{i_1} & Z_2^{i_1} & \ldots & Z_\nu^{i_1} \\ Z_1^{i_2} & Z_2^{i_2} & \ldots & Z_\nu^{i_2} \\ \vdots & \vdots & \ddots & \vdots \\ Z_1^{i_\nu} & Z_2^{i_\nu} & \ldots & Z_\nu^{i_\nu} \\ Z_1^{i_{\nu+1}} & Z_1^{i_{\nu+1}} & \ldots & Z_\nu^{i_{\nu+1}} \end{pmatrix}$$

and

$$X(J) = \begin{pmatrix} Z_1^{j_1} & Z_2^{j_1} & \dots & Z_\nu^{j_1} \\ Z_1^{j_2} & Z_2^{j_2} & \dots & Z_\nu^{j_2} \\ \vdots & \vdots & \ddots & \vdots \\ Z_1^{j_\nu} & Z_2^{j_\nu} & \dots & Z_\nu^{j_\nu} \\ Z_1^{j_{\nu+1}} & Z_1^{j_{\nu+1}} & \dots & Z_\nu^{j_{\nu+1}} \end{pmatrix} \tag{7}$$

See [11, 12]. Next, consider the matrix $S(I, J)$ of size $(\nu + 1) \times (\nu + 1)$ as follows:

$$S(I, J) = X(I)X(J)^T \tag{8}$$

where $X(J)^T$ denotes the transpose of the matrix $X(J)$. Then

$$S(I, J) = \begin{pmatrix} S_{i_1+j_1} & S_{i_1+j_2} & \dots & S_{i_1+j_{\nu+1}} \\ S_{i_2+j_1} & S_{i_2+j_2} & \dots & S_{i_2+j_{\nu+1}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i_\nu+j_1} & S_{i_\nu+j_2} & \dots & S_{i_\nu+j_{\nu+1}} \\ S_{i_{\nu+1}+j_1} & S_{i_{\nu+1}+j_2} & \dots & S_{i_{\nu+1}+j_{\nu+1}} \end{pmatrix} \tag{9}$$

where the summation of the subindices of the S_i's is modulo n , and $det(S(I, J)) = 0$. If there is only one unknown syndrome, say S_r among the entries of $S(I, J)$, then S_r can be expressed as a function in terms of some known syndromes. Hence, during the decoding process, one is able to calculate the value of S_r with the information about those known syndromes. The detailed procedures to determine the primary unknown syndromes.

Theorem 3.1.1. *[2] If among the entries of $S(I, J)$, there is only one unknown syndrome, say S_r, then S_r can be expressed as the ratio of two determinants of matrices obtained from $S(I, J)$. If S_r appears in the $(i, j)^{th}$ position of $S(I, J)$, then*

$$S_r = \frac{det(\Delta_0)}{det(\Delta)}$$

provided that $det(\Delta) \neq 0$, where Δ_0 is the $(\nu + 1) \times (\nu + 1)$ matrix that is identical to $S(I, J)$, except for the $(i, j)^{th}$ entrywhich equals 0 instead of S_r, and Δ is the $\nu \times \nu$ submatrix of $S(I, J)$ obtained by deleting the i^{th} row and j^{th} column of $S(I, J)$.

For an arbitrary binary QR code the following theorem, proved in Reference[6], holds.

Theorem 3.1.2. *[5] Let (n, k, d) be a binary QR code with Q, the set of quadratic residues modulo n, and the base set S. Then the error pattern has a weight of at most one if and only if the following two conditions hold:*

(i) $S_i = S_1^i \quad$ *for all* $i \in S$

(ii) $S_1^{n+1} = S_1$

A necessary and sufficient condition for exactly two errors to occur in the received codeword for the subclass of the QR codes of length $n = 8l + 1$ is found in the next theorem.

Theorem 3.1.3. *[5] Let* (n, k, d) *be a binary QR code of length* $n = 8l + 1$ *and quadratic residue set Q. Then, the error pattern has a weight of two if and only if the following conditions are satisfied:*

(i) $Tr(\frac{1}{S_1 S_{-1}}) = 0$ *where* $Tr(.)$ *denote the trace*

(ii) $S_i = Z_1^i + Z_2^i$, $\forall\ i \in S$ *where* Z_1 *and* Z_2 *are the roots of the quadratic equation,* $z^2 + S_1 z + \frac{S_1}{S_{-1}} = 0$

Note that: The trace of k in $GF(2^m)$ is defined by $tr(k) = \sum_{i=0}^{m-1} k^{2^i}$.*See*[6].

3.2 Decoding the (41, 21, 9) quadratic residue code

The algebraic decoding method for $(41, 21, 9)$ *QR code was given in [6], [13]. In this section we will give a new algebraic decoding method for* $(41, 21, 9)$ *QR code.*

The class of quadratic residue QR codes was introduced by Prange in 1958 [17]. It is a nice family of cyclic codes and has approximately 1/2 code rates. In the past decades, most of decoding methods for the QR codes are used to solve the Newton identities that are nonlinear, multivariante equations of quite high degree. It becomes very difficult when the weight of the occurred error becomes large. Moreover, different QR codes use different sets of conditions to determine the error locations. As a result, it is not practical for software implementation.

For the (41, 21, 9) QR code, the set of quadratic residue modulo 41, is

$$Q_{41} = \{1, 2, 4, 5, 8, 9, 10, 16, 18, 20, 21, 23, 25, 31, 32, 33, 36, 37, 39, 40\}.$$

The smallest extension field of F_2 is $F_{2^{20}}$ as $2^{20} \equiv 1 \bmod 41$. let α be a root of the primitive polynomial $f(x) = x^{20} + x^3 + 1$ which is irreducible over F_2 and let $\beta = \alpha^{(2^m-1)/n} = \alpha^{(2^{20}-1)/41} = \alpha^{25575}$ be a primitive 41st root of unity in $GF(2^{20})$. The generator polynomial of binary (41,21, 9) QR code can be found by **Maple 13** as follows

$$alias(\alpha = RootOf(x^{20}+x^3+1)); Expand((x-\alpha^{25575^1})\cdot(x-\alpha^{25575^2})\ldots(x-\alpha^{25575^{39}})\cdot(x-\alpha^{25575^{40}}))mod2.$$

to get

$$g(x) = \prod_{i\in Q_{41}}(x-\beta^i) = \prod_{i\in Q_{41}}(x-\alpha^{25575^i}) = x^{20}+x^{19}+x^{17}+x^{16}+x^{14}+x^{11}+x^{10}+x^9+x^6+x^4+x^3+x+1.$$

Let a codeword $c(x) = m(x)g(x)$ be transmitted through a noisy channel to obtain a received codeword of the form $r(x) = c(x) + e(x) = \sum_{i=1}^{40} c_i x^i + \sum_{i=1}^{40} e_i x^i$, where e(x) is the polynomial of the received error pattern vector.

★ New algebraic decoding:

The base set of the (41, 21, 9) QR code is $S = \{1\}$, a singleton set. Every known syndromes (respectively, unknown syndromes) can be expressed as some power of S_1

(resp.,S_3) as follows $S_2 = S_1^2$, $S_4 = S_1^4$, $S_5 = S_1^{128}$, $S_8 = S_1^8$, $S_9 = S_1^{32768}$, $S_{10} = S_1^{256}$, $S_{16} = S_1^{16}$, $S_{20} = S_1^{512}$, $S_{21} = S_1^{524288}$, $S_{23} = S_1^{64}$, $S_{25} = S_1^{16384}$, $S_{31} = S_1^{262144}$, $S_{32} = S_1^{32}$, $s_{33} = S_1^{8192}$, $S_{36} = S_1^{131072}$, $S_{37} = S_1^{4096}$, $S_{39} = S_1^{2048}$, $S_{-1} = S_{40} = S_1^{1024}$ and $S_{38} = S_3^{1024}$ see[13]. Since $t = \frac{(d-1)}{2} = 4$, this code allows for the correction of up to four errors. In other words, there are five cases to be considered which are:

case 0: No error in the received codeword if and only if $S_1 = 0$; otherwise go to Case 1.

case 1: One error in the received codeword if and only if the two conditions in Theorem 3.1.2 holds, i.e. $S_1^{41} = 1$; otherwise go to Case 2.

case 2: Two errors in the received codeword if and only if the two conditions in Theorem 3.1.3 holds. From Theorem 3.1.2 and Theorem 3.1.3 one has that,

$$S_1 + \sigma_1 = 0 \tag{10}$$

$$S_{-1} + \frac{\sigma_1}{\sigma_2} = 0 \tag{11}$$

which, when solved, yields

$$\sigma_1 = S_1$$

$$\sigma_2 = \frac{S_1}{S_{-1}} = \frac{S_1}{S_{40}}$$

The error-locator polynomial is obtained from these $\sigma_k s$ as follows $L_2(z) = z^2 + \sigma_1 z + \sigma_2 = 0$, if $L_2(z)$ has the roots Z_1 and Z_2 with $Z_i^{41} = 1$ (i=1, 2), then two errors have occurred ; otherwise go to Case 3.

case 3: For three errors, one has

$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 + \sigma_3 = 0 \tag{12}$$

$$S_{-1} + \frac{\sigma_2}{\sigma_3} = 0 \tag{13}$$

$$S_{-3} + \frac{\sigma_2}{\sigma_3} S_{-2} + \frac{\sigma_1}{\sigma_2} S_{-1} + \frac{1}{\sigma_3} = 0 \tag{14}$$

The coefficients of the error-locator polynomial, found by solving equations 10, 12 and 13 are

$$\sigma_1 = S_1$$
$$\sigma_2 = \frac{S_{-1}(S_3 + S_1^3)}{S_1 S_{-1} + 1}$$
$$\sigma_3 = \frac{S_3 + S_1^3}{S_1 S_{-1} + 1} \tag{15}$$

In this case the unknown syndrome S_3 can be determined by choosing $I_3 = \{0, 31, 39, 40\}$ and $J_3 = \{0, 1, 2, 5\}$, yields the matrix $S(I_3, J_3)$ of size 4×4 as follows

$$\begin{bmatrix} S_0 & S_1 & S_2 & S_5 \\ S_{31} & S_{32} & S_{33} & S_{36} \\ S_{39} & S_{40} & S_0 & S_3 \\ S_{40} & S_0 & S_1 & S_4 \end{bmatrix} = \begin{bmatrix} 1 & S_1 & S_1^2 & S_1^{128} \\ S_1^{262144} & S_1^{32} & S_1^{8192} & S_1^{131072} \\ S_1^{2048} & S_1^{1024} & 1 & S_3 \\ S_1^{1024} & 1 & S_1 & S_1^4 \end{bmatrix}$$

See [13], which satisfied $det(S(I_3, J_3)) = 0$. when solved, yields
$S_3 = \frac{S_1^3(S_1^{133086}+S_1^{250}+S_1^{373}+S_1^{263261}+S_1^{1148}+S_1^{9184}+S_1^{8409}+S_1^{8532}+S_1^{262113}+S_1^{10332}+S_1^{131036}+S_1^{262236}+S_1^{263138}+1)}{S_1^{9184}+S_1^{8159}+S_1^{1025}+1}$
Next, it is necessary to find the condition for the case of exactly three errors with the known $\sigma_i (i = 1, 2, 3)$ in equation 15 . Given

$$S_1^2 S_{-1}^2 + 1 = (S_{-3} + S_{-1}^3)(S_3 + S_1^3) \tag{16}$$

If the conditions for case 0, case 1 and case 2 do not hold, equation 16 is also a sufficient condition for the case of exactly three errors.
So we have

$$L_3(z) = z^3 + \sigma_1 z^2 + \sigma_2 z + \sigma_3 = 0$$
$$L_3(z) = z^3 + S_1 z^2 + [\frac{S_{-1}(S_3 + S_1^3)}{S_1 S_{-1} + 1}]z + (\frac{S_3 + S_1^3}{S_1 S_{-1} + 1}) = 0$$

Furthermore the error pattern is found by a search of the roots of $L_3(z)$ over the n^{th} which found by **Maple 13** or **Magma**.

case 4: For four errors, one has

$$S_{-1} + \frac{\sigma_3}{\sigma_4} = 0 \tag{17}$$

$$S_{-3} + \frac{\sigma_3}{\sigma_4} S_{-2} + \frac{\sigma_2}{\sigma_4} S_{-1} + \frac{\sigma_1}{\sigma_4} = 0 \tag{18}$$

and

$$S_{-5} + \frac{\sigma_3}{\sigma_5} S_{-3} + \frac{\sigma_2}{\sigma_5} S_{-2} + \frac{\sigma_1}{\sigma_5} S_{-1} = 0 \tag{19}$$

which, when solved, yields

$$\sigma_1 = S_1$$

$$\sigma_2 = \frac{S_3 S_{38} + S_3 S_{39} S_{40} + S_1^3 S_{38} + S_1^3 S_{39} S_{40} + S_{40} S_1}{S_{38} S_1 + S_{39} S_{40} S_1 + S_{40}^2}$$

$$\sigma_3 = \frac{S_{40}(S_{40} S_3 + S_{40} S_1^3 + S_1^2)}{S_{38} S_1 + S_{39} S_{40} S_1 + S_{40}^2}$$

$$\sigma_4 = \frac{S_{40} S_3 + S_{40} S_1^3 + S_1^2}{S_{38} S_1 + S_{39} S_{40} S_1 + S_{40}^2} \tag{20}$$

Here, let $I_4 = \{0, 2, 8, 9, 20\}$ and $J_4 = \{0, 1, 23, 31, 37\}$, to have the matrix $S(I_4, J_4)$ of size 5×5 as follows

$$\begin{bmatrix} S_0 & S_1 & S_{23} & S_{31} & S_{37} \\ S_2 & S_3 & S_{25} & S_{33} & S_{39} \\ S_8 & S_9 & S_{31} & S_{39} & S_4 \\ S_9 & S_{10} & S_{32} & S_{40} & S_5 \\ S_{20} & S_{21} & S_2 & S_{10} & S_{16} \end{bmatrix} = \begin{bmatrix} 0 & S_1 & S_1^{64} & S_1^{262144} & S_1^{4096} \\ S_1^2 & S_3 & S_1^{16384} & S_1^{8192} & S_1^{2048} \\ S_1^8 & S_1^{32768} & S_1^{262144} & S_1^{2048} & S_1^4 \\ S_1^{32768} & S_1^{256} & S_1^{32} & S_1^{1024} & S_1^{128} \\ S_1^{512} & S_1^{524288} & S_1^2 & S_1^{256} & S_1^{16} \end{bmatrix}.$$

determine S_3 by solving $det(S(I_4, J_4)) = 0$, to have

$S_3 = (1 + S_1^{2050} + S_1^{287} + S_1^{738} + S_1^{1804} + S_1^{1886} + S_1^{2091} + S_1^{2337} + S_1^{2788} + S_1^{4346} + S_1^{4633} +$
$S_1^{6109} + S_1^{7954} + S_1^{8036} + S_1^{8241} + S_1^{8446} + S_1^{8733} + S_1^{12259} + S_1^{16482} + S_1^{17138} + S_1^{17630} +$
$S_1^{18778} + S_1^{20705} + S_1^{23001} + S_1^{32923} + S_1^{33579} + S_1^{36121} + S_1^{36572} + S_1^{36859} + S_1^{37597} + S_1^{40672} +$
$S_1^{41369} + S_1^{45305} + S_1^{49118} + S_1^{50922} + S_1^{54489} + S_1^{67609} + S_1^{73513} + S_1^{77531} + S_1^{85977} + S_1^{262133} +$
$S_1^{262236} + S_1^{262892} + S_1^{264163} + S_1^{265434} + S_1^{266459} + S_1^{270682} + S_1^{274905} + S_1^{278513} + S_1^{279005} +$

$S_1^{294667} + S_1^{294749} + S_1^{296922} + S_1^{297209} + S_1^{302826} + S_1^{311641} + S_1^{329435} + S_1^{343785} + S_1^{524267} +$
$S_1^{525087} + S_1^{526235} + S_1^{526809} + S_1^{527137} + S_1^{530171} + S_1^{532385} + S_1^{536321} + S_1^{545505} + S_1^{560921} +$
$S_1^{565021} + S_1^{579289} + S_1^{786175} + S_1^{788225} + S_1^{791259} + S_1^{802657} + S_1^{831193} + S_1^{835293} + S_1^{1048411} +$
$S_1^{1083097})/S_1^{161}(1 + S_1^{656} + S_1^{1148} + S_1^{2296} + S_1^{3936} + S_1^{4674} + S_1^{6236} + S_1^{32636} + S_1^{34440} +$
$S_1^{38458} + S_1^{261744} + S_1^{261826} + S_1^{262236} + S_1^{267320} + S_1^{294462} + S_1^{298808} + S_1^{524472} + S_1^{556616})$

Next, it is necessary to find the condition for the case of exactly four errors with the known σ_i (i = 1, 2, 3, 4) in equation 19. Gives

$$\frac{(S_3+S_1^3)(S_{37}S_{40}+S_{38}S_{40}^2)+S_3S_{39}(S_{38}+S_{39}S_{40})+S_1^3S_{39}(1+S_{39}S_{40})+S_1S_{39}S_{40}(1+S_1S_{40})+S_1(S_{40}^2+S_1S_{37})}{S_{40}(S_3+S_1^3)+S_1^2} = 0 \quad (21)$$

A full computer search shows that, in the case of exactly four errors, $S_{40}(S_3+S_1^3)+S_1^2$ is always nonzero. If the conditions for case 0, case 1, case 2 and case 3 do not hold, equation 21 is also a sufficient condition for the case of exactly four errors.

So we have

$$L_4(z) = z^4 + \sigma_1 z^3 + +\sigma_2 z^2 \sigma_3 z + \sigma_4 = 0$$

$L_4(z) = z^4 + S_1 z^3 + \frac{S_3S_{38}+S_3S_{39}S_{40}+S_1^3S_{38}+S_1^3S_{39}S_{40}+S_{40}S_1}{S_{38}S_1+S_{39}S_{40}S_1+S_{40}^2} z^2 + \frac{S_{40}(S_{40}S_3+S_{40}S_1^3+S_1^2)}{S_{38}S_1+S_{39}S_{40}S_1+S_{40}^2} z +$
$\frac{S_{40}S_3+S_{40}S_1^3+S_1^2}{S_{38}S_1+S_{39}S_{40}S_1+S_{40}^2} = 0$

Furthermore the error pattern is found by a search of the roots of $L_4(z)$ over the n^{th} which found by **Maple 13** or **Magma**. The flowchart of the entire process for decoding this code up to four errors is depicted in the flow diagram in Figure 1

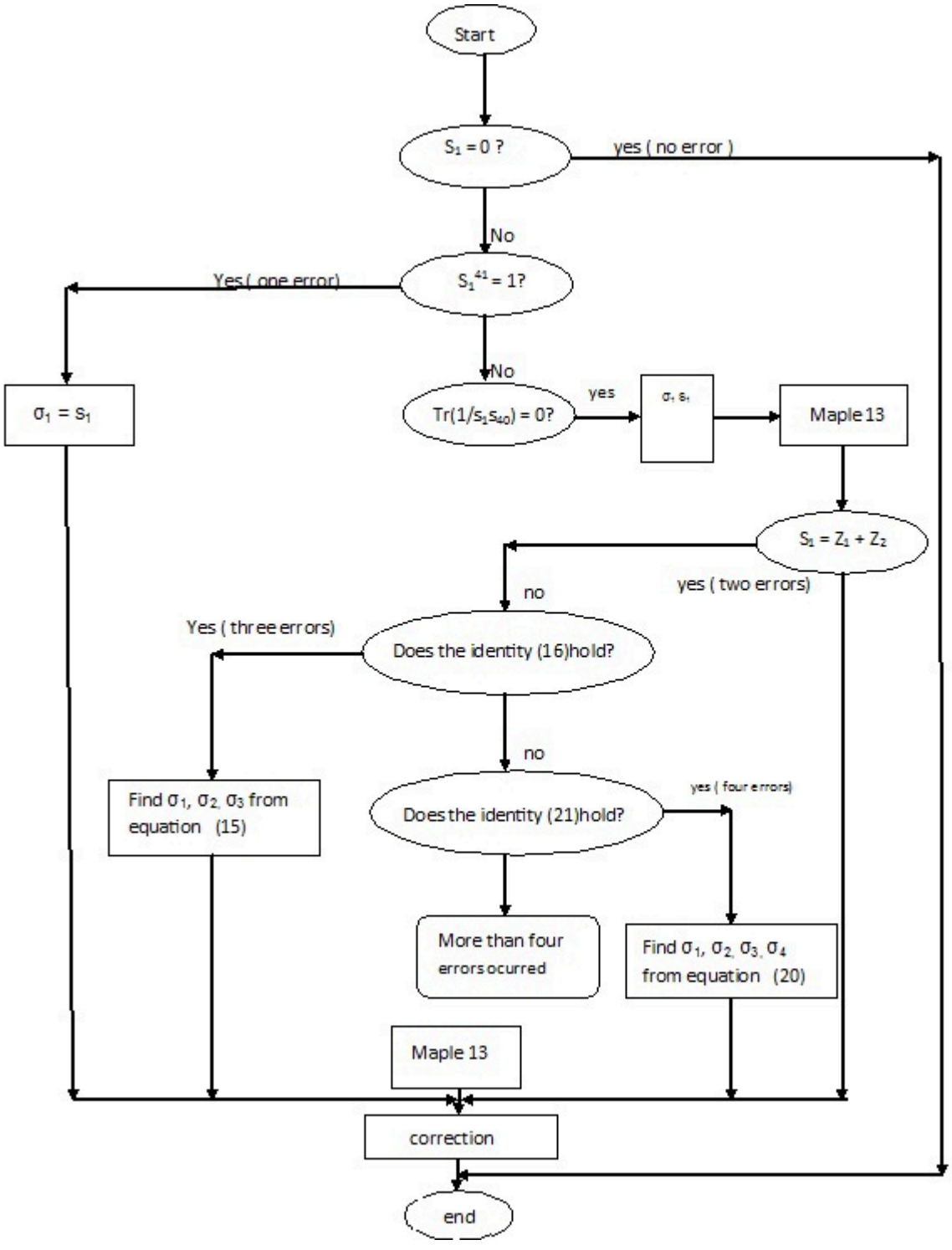

Figure 3.1: Flowchart of the algorithm for decoding the (41,21, 9) QR code

3.3 Decoding the (73, 37, 13) quadratic residue code

Let $n = p = 73$ be odd prime where $n = p = 73 \equiv 1 \bmod 73$. Let p be odd prime and let a be in $\mathbb{Z}_p$ with $a \equiv 0 \bmod p$. Then a is a square if and only if $a^{\frac{p-1}{2}} \equiv 1 \bmod p$, So For the $(73, 37, 13)$ QR code, the set of quadratic residue modulo 73, is

$Q_{73} = \{$ 1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 19, 23, 24, 25, 27, 32, 35, 36, 37, 38, 41, 46, 48, 49, 50, 54, 55, 57, 61, 64 , 65, 67, 69, 70, 71, 72$\}$.

Note that we found Q_{73} by **Maple 13**; For example 8 is nonzero quadratic residues mod 73 since $(8^{36} - 1) =$ **zero** mod 73, but 52 is not nonzero quadratic residues mod 73 since $(52^{36} - 1) \neq$ **zero** mod 73 which written in **Maple 13** as:

$$(8^{36} - 1) \ \ mod \ \ 73 \qquad to \ \ get \ \ the \ \ result \ \ 0$$

.

Since $2^9 \equiv 1 \bmod 73$, the smallest extension field of $\mathbb{F}_2$ is $\mathbb{F}_{2^9} = GF(2^9)$. Let $f(x) = x^9 + x^5 + x^4 + x + 1$ which is irreducible over $\mathbb{F}_2$ we can check this by **Maple 13** as follows:

$$Factors(x^9 + x^5 + x^4 + x + 1) \quad mod \quad 2$$

To get the result $[1, [[x^9 + x^5 + x^4 + x + 1, 1]]]$, i.e $f(x)$ is irreducible polynomial over $\mathbb{F}_2$. and let $\beta = \alpha^{(2^m-1)/n} = \alpha^{(2^9-1)/73} = \alpha^7$ be a primitive 73^{th} root of unity in $GF(2^9)$. The generator polynomial of binary (73, 37, 13) QR code can be found by **Maple 13** as follows:

$alias(\alpha = RootOf(x^9+x^5+x^4+x+1)); Expand((x-\alpha^{7^1})\cdot(x-\alpha^{7^2})\ldots(x-\alpha^{7^{71}})\cdot(x-\alpha^{7^{72}}))$ mod 2.

to get $g(x) = \prod_{i \in Q_{73}}(x - \beta^i) = \prod_{i \in Q_{73}}(x - \alpha^{7^i}) = (1 + x + x^9)(1 + x + x^2 + x^4 + x^9)(1 + x + x^3 + x^6 + x^9)(1 + x^2 + x^5 + x^6 + x^9)$.

The base set of the (73, 37, 13) QR code is $S = \{1, 3, 9, 25\}$. Since $t = \frac{(d-1)}{2} = 6$, this code allows for the correction of up to sex errors. **We will study up to four error**

case 0: No error in the received codeword if and only if $S_1 = 0$; otherwise go to Case 1.

case 1: One error in the received codeword if and only if the two conditions in Theorem 3.1.2 holds, i.e. $S_3 = S_1^3$, $S_9 = S_1^9$, $S_{25} = S_1^{25}$ and $S_1^{73} = 1$; otherwise go to Case 2.

case 2: Two errors in the received codeword if and only if the two conditions in Theorem 3.1.3 holds. From Newton identities in equation (5) and (6) we have

$$S_1 + \sigma_1 = 0 \text{.....(1)}$$

$$S_{-1} + \frac{\sigma_1}{\sigma_2} = 0 \text{.....(2)}$$

which, when solved, yields

$$\sigma_1 = S_1$$
$$\sigma_2 = \frac{S_1}{S_{-1}} = \frac{S_1}{S_{72}}$$

The error-locator polynomial is obtained from these $\sigma_k s$ as follows $L_2(z) = z^2 + \sigma_1 z + \sigma_2 = 0$, if $L_2(z)$ has the roots Z_1 and Z_2 with $Z_i^{73} = 1$ (i=1, 2), then two errors have occurred ; otherwise go to Case 3.

case 3: For three errors, from Newton identities in equation (5) and (6) we have

$$S_1 + \sigma_1 = 0$$

$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 + \sigma_3 = 0 \text{.....(3)}$$

$$S_{-1} + \frac{\sigma_2}{\sigma_3} = 0 \text{.....(4)}$$

$$S_{-3} + \frac{\sigma_2}{\sigma_3} S_{-2} + \frac{\sigma_1}{\sigma_2} S_{-1} + \frac{1}{\sigma_3} = 0 \text{.....(5)}$$

Now we can find σ_1, σ_2, σ_3 by the known syndromes

$$S_1 = \sigma_1$$

$\star$ **To find** σ_2: we have $S_{-1} + \frac{\sigma_2}{\sigma_3} = 0 \Longrightarrow \sigma_3 = \frac{\sigma_2}{S_{-1}}$

From equation (3) we have $(S_3 + S_1^3) + \sigma_2 S_1 + \frac{\sigma_2}{S_{-1}} = 0 \Longrightarrow (S_3 + S_1^3) \cdot S_{-1} + \sigma_2 S_1 S_{-1} + \sigma_2 = 0 \Longrightarrow (S_3 + S_1^3) S_{-1} = \sigma_2 (S_1 S_{-1} + 1) \Longrightarrow \sigma_2 = \frac{(S_3 + S_1^3) \cdot S_{-1}}{(S_1 S_{-1} + 1)}$ .....(6)

$\star$ **To find** σ_3: From equation (3) we have $\sigma_2 = S_{-1}\sigma_3$, So
$S_3 + S_1^3 + S_{-1}S_1\sigma_3 + \sigma_3 = 0 \Longrightarrow S_3 + S_1^3 + \sigma_3(S_{-1}S_1 + 1) = 0 \Longrightarrow \sigma_3 = \frac{S_3+S_1^3}{S_{-1}S_1+1}$.
Now the important condition for $\nu = 3$ can be computed by substitute σ_1, σ_2, σ_3 in the equation (5)

$$S_{-3} + \frac{(S_3 + S_1^3)S_{-1}}{S_{-1}S_1 + 1} \times \frac{S_{-1}S_1 + 1}{S_3 + S_1^3}S_{-2} + \frac{S_1}{\frac{(S_3+S_1^3)}{S_{-1}S_1+1}}S_{-1} + \frac{S_{-1}S_1 + 1}{S_3 + S_1^3} = 0$$

$$S_{-3} + S_{-1}S_{-2} + \frac{S_1S_{-1}(S_{-1}S_1 + 1)}{S_3 + S_1^3} + \frac{S_{-1}S_1 + 1}{S_3 + S_1^3} = 0$$

$$(S_{-3} + S_{-1}^3)(S_3 + S_1^3) + S_1^2S_{-1}^2 + S_1S_{-1} + S_1S_{-1} + 1 = 0$$

$$(S_{-3} + S_{-1}^3)(S_3 + S_1^3) = S_1^2S_{-1}^2 + 1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (*)$$

If the conditions for case 0, case 1 and case 2 do not hold, eqn. * is also a sufficient condition for the case of exactly three errors.
Note that: the result in equations (5) can be found directly by **Maple 13** as follows:

$$solve(s1 + \sigma1 = 0, s(-1) + \sigma2 \cdot \sigma3 = 0, s3 + s1^3 + \sigma2 \cdot s1 + \sigma3 = 0, [\sigma1, \sigma2, \sigma3])\ mod\ 2$$

To get the result :

$$[[\sigma1 = s1, \sigma2 = s(-1) \cdot (s3 + s1^3)/(s(-1) \cdots 1 + 1), \sigma3 = (s3 + s1^3)/(s(-1) \cdots 1 + 1)]]$$

case 4: For four errors, equation (1) and (3) still hold if $\nu = 4$. From Newton identities instead of equation (5) and (6), we have

$$S_{-1} + \frac{\sigma_3}{\sigma_4} = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

$$S_{-3} + \frac{\sigma_3}{\sigma_4}S_{-2} + \frac{\sigma_2}{\sigma_4}S_{-1} + \frac{\sigma_1}{\sigma_4} = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

from equation (7) we have

$$S_{-3}\sigma_4 + \sigma_3 S_{-2} + \sigma_2 S_{-1} + S_1 = 0$$

Substitute $\sigma_4 = \frac{\sigma_3}{S_{-1}}$ in equation (8) to get

$$\frac{S_{-3}\sigma_3}{S_{-1}} + \sigma_3 S_{-2} + \sigma_2 S_{-1} + S_1 = 0$$

$\star$ **To find** σ_2: $\sigma_3 = (S_3 + S_1^3) + \sigma_2 S_1$

$$\frac{S_{-3}}{S_{-1}}[(S_3 + S_1^3) + \sigma_2 S_1] + [(S_3 + S_1^3) + \sigma_2 S_1]S_{-2} + \sigma_2 S_{-1} + S_1 = 0$$

$$\frac{S_{-3}}{S_{-1}}(S_3 + S_1^3) + \frac{\sigma_2 S_1 S_{-3}}{S_{-1}} + (S_3 + S_1^3)S_{-2} + \sigma_2 S_1 S_{-2} + \sigma_2 S_{-1} + S_1 = 0$$

$$\sigma_2[\frac{S_1 S_{-3}}{S_{-1}} + S_1 S_{-2} + S_{-1}] = (S_3 + S_1^3)(\frac{S_{-3}}{S_{-1}} + S_{-2}) + S_1$$

$$\sigma_2[\frac{S_1 S_{-3} + S_1 S_{-1} S_{-2} + S_{-1}^2}{S_{-1}}] = (S_3 + S_1^3)(\frac{S_{-3} + S_{-1} S_{-2}}{S_{-1}}) + S_1$$

$$\sigma_2(S_1 S_{-3} + S_1 S_{-1} S_{-2} + S_{-1}^2) = (S_3 + S_1^3)(S_{-3} + S_{-1}^3) + S_1 S_{-1}$$

$$\sigma_2[S_1(S_{-3} + S_{-1}^3) + S_{-1}^2] = (S_3 + S_1^3)(S_{-3} + S_{-1}^3) + S_1 S_{-1}$$

$$\sigma_2 = \frac{(S_3 + S_1^3)(S_{-3} + S_{-1}^3) + S_1 S_{-1}}{S_1(S_{-3} + S_{-1}^3) + S_{-1}^2}.$$

$\star$ **To find** σ_3:We have $\sigma_2 = \frac{(S_3 + S_1^3) + \sigma_3}{S_1}$

$$\frac{S_{-3}\sigma_3}{S_{-1}} + \sigma_3 S_{-2} + \frac{[(S_3 + S_1^3) + \sigma_3]S_{-1}}{S_1} + S_1 = 0$$

$$\frac{S_{-3}\sigma_3}{S_{-1}} + \sigma_3 S_{-2} + \frac{(S_3 + S_1^3)S_{-1} + \sigma_3 S_{-1}}{S_1} + S_1 = 0$$

$$S_1 S_{-3}\sigma_3 + S_1 S_{-1} S_{-2}\sigma_3 + (S_3 + S_1^3)S_{-1}^2 + \sigma_3 S_{-1}^2 + S_1^2 S_{-1} = 0$$

$$\sigma_3(S_1 S_{-3} + S_1 S_{-1}^3 + S_{-1}^2) = (S_3 + S_1^3)S_{-1}^2 + S_1^2 S_{-1}$$

$$\sigma_3[S_1(S_{-3} + S_{-1}^3) + S_{-1}^2] = (S_3 + S_1^3)S_{-1}^2 + S_1^2 S_{-1}$$

$$\sigma_3 = \frac{S_1^2 S_{-1} + S_{-1}^2(S_3 + S_1^3)}{S_1(S_{-3} + S_{-1}^3) + S_{-1}^2}.$$

Finally to find σ_4:We have $\sigma_4 = \frac{\sigma_3}{S_{-1}}$

$$\therefore \sigma_4 = \frac{\frac{S_1^2 S_{-1}}{S_{-1}} + \frac{S_{-1}^2(S_3+S_1^3)}{S_{-1}}}{S_1(S_{-3}+S_{-1}^3)+S_{-1}^2}$$
$$= \frac{S_1^2 + S_{-1}(S_3+S_1^3)}{S_1(S_{-3}+S_{-1}^3)+S_{-1}^2}.$$

Now the important condition for $\nu = 4$ can be computed by substitute σ_1, σ_2, σ_3 and σ_4 in the equation (8)

$$S_{-3} + \frac{\sigma_3}{\sigma_4}S_{-2} + \frac{\sigma_2}{\sigma_4}S_{-1} + \frac{\sigma_1}{\sigma_4} = 0$$

$$S_{-3} + \frac{[S_1^2 S_{-1} + S_{-1}^2(S_3+S_1^3)]S_{-2}}{S_1^2 + S_{-1}(S_3+S_1^3)} + \frac{[S_1 S_{-1} + (S_3+S_1^3)(S_{-3}+S_{-1}^3)]S_{-1}}{S_1^2 + S_{-1}(S_3+S_1^3)} + \frac{S_1}{S_1^2 + S_{-1}(S_3+S_1^3)} = 0$$

$$S_1^2 S_{-3} + S_{-1}S_{-3}(S_3+S_1^3) + S_1^2 S_{-1}S_{-2} + S_{-1}^2 S_{-2}(S_3+S_1^3) + S_1 S_{-1}^2 + (S_3+S_1^3)(S_{-3}+S_{-1}^3)S_{-1} + S_1 = 0$$

$$(S_3+S_1^3)(S_{-1}S_{-3} + S_{-1}^2 S_{-2} + S_{-1}S_{-3} + S_{-1}S_{-1}^3) + S_1^2(S_{-3}+S_{-1}^3) + S_1(S_{-1}^2+1) = 0$$

$$S_1^2(S_{-3}+S_{-1}^3) + S_1(S_{-1}^2+1) = 0$$

$$S_1(S_1 S_{-3} + S_1 S_{-1}^3 + S_{-1}^2 + 1) = 0$$

$$S_1 S_{-3} + S_1 S_{-1}^3 + S_{-1}^2 + 1 = 0 \qquad \text{..}(**)$$

If the conditions for case 0, case 1, case 2 and case 3 do not hold, eqn. ** is also a suficient condition for the case of exactly four errors.

3.4 Decoding the (47, 24, 11) quadratic residue code

In this section we will give a different algebraic decoding which studied in previous sections. This code correct up to five errors.

The length of this code is a prime number of the form $n = 8m - 1$, where m is some integer. The set of quadratic residue modulo 47, is

Q_{47} = { 1, 2, 3, 4, 6, 7, 8, 9, 12, 14, 16, 17, 18,21, 24, 25, 27, 28, 32, 34, 36, 37, 42}. Since $t = \frac{d-1}{2} = 5$, the (47, 24, 11) QR code allows for the correction of up to five errors. In other words, there are six cases to be considered as follows:

Case 0: No error in the received codeword if and only if $S_1 = 0$; otherwise go to case 1.

Case 1: One error in the received codeword if and only if $S_1^{47} = 1$; otherwise go to case 2.

Case 2: For two errors in the received codeword we have

$$S_1 + \sigma_1 = 0 \text{.....(1)}$$

$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 = 0 \text{.....(2)}$$

From equation (1) we have $\sigma_1 = S_1$

From equation (1) we have $\sigma_2 = \frac{S_3 + S_1^3}{S_1}$.

Thus, the error-locator polynomial

$$L_2(z) = z^2 + \sigma_1 z + \sigma_2 = 0$$

$$L_2(z) = z^2 + S_1 z + \frac{S_3 + S_1^3}{S_1} = 0$$

• If this equation has two roots Z_1 and Z_2, and $Z_1^{47} = Z_2^{47} = 1$, then two errors occur; otherwise more errors occur, and then go to case 3.

Case 3: For three errors, we have the equations

$$S_1 + \sigma_1 = 0$$

$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 + \sigma_3 = 0 \text{.....(3)}$$

$$S_9 + \sigma_1 S_8 + \sigma_2 S_7 + \sigma_3 S_6 = 0 \text{.....(4)}$$

⋆ **To find** σ_2**:**From equation (3) we have $\sigma_3 = (S_3 + S_1^3) + \sigma_2 S_1$
Substitute in equation (4) with $S_8 = S_{1\times 2^3} = S_1^8$

$$(S_9 + S_1^9) + \sigma_2 S_7 + [(S_3 + S_1^3) + \sigma_2 S_1]S_6 = 0$$

$$(S_9 + S_1^9) + \sigma_2 S_7 + (S_3 + S_1^3)S_6 + \sigma_2 S_1 S_6 = 0$$

$$\sigma_2 = \frac{(S_9 + S_1^9) + (S_3 + S_1^3)S_6}{S_1 S_6 + S_7}.$$

⋆ **To find** σ_3**:**

$$\sigma_3 = (S_3 + S_1^3) + \frac{(S_9 + S_1^9)S_1 + (S_3 + S_1^3)S_1 S_6}{S_1 S_6 + S_7}.$$

• If $S_1 S_6 + S_7 \neq 0$, then there are exactly three errors, otherwise go to case 4.

Case 4: Four error, in this case there are not enough consecutive syndromes, for this we look to the first unknown syndrome which is S_5. Choose $I = \{1, 3, 4, 14, 21\}$ and $J = \{0, 3, 4, 13, 33\}$ to get the matrix

$$S(I,J) = \begin{bmatrix} S_1 & S_3 & S_4 & S_{14} & S_{21} \\ S_4 & S_6 & S_7 & S_{17} & S_{24} \\ S_5 & S_7 & S_8 & S_{18} & S_{25} \\ S_{14} & S_{16} & S_{17} & S_{27} & S_{34} \\ S_{34} & S_{36} & S_{37} & S_0 & S_7 \end{bmatrix},$$

Here, $S_0 = 0$ since $\nu = 4$, also we have that in this matrix all syndromes are known except S_5 and $det(S(I,J)) = 0$ since the $rank(S(I,J)) \leq \nu$ By Theorem 3.1.1 we have

$$S_5 = \frac{det(\Delta_0)}{det(\Delta)}$$

where

$$\Delta_0 = \begin{bmatrix} S_1 & S_3 & S_4 & S_{14} & S_{21} \\ S_4 & S_6 & S_7 & S_{17} & S_{24} \\ S_5 & S_7 & S_8 & S_{18} & S_{25} \\ S_{14} & S_{16} & S_{17} & S_{27} & S_{34} \\ S_{34} & S_{36} & S_{37} & S_0 & S_7 \end{bmatrix},$$

$$\Delta = \begin{bmatrix} S_3 & S_4 & S_{14} & S_{21} \\ S_6 & S_7 & S_{17} & S_{24} \\ S_{16} & S_{17} & S_{27} & S_{34} \\ S_{36} & S_{37} & S_0 & S_7 \end{bmatrix}.$$

- If $det(\Delta) \neq 0$, then four error occur, otherwise go to case 5.

Case 5: For five errors, this is a hard case, since we cant contain matrix with one unknown syndrome S_5, for this choose $I_1 = \{0, 1, 4, 8, 12, 32\}$ and $J_1 = \{0, 2, 4, 16, 20, 24\}$ to obtain the matrix

$$S(I, J) = \begin{bmatrix} S_0 & S_1 & S_4 & S_8 & S_{12} & S_{32} \\ S_2 & S_3 & S_6 & S_{10} & S_{14} & S_{34} \\ S_4 & S_5 & S_8 & S_{12} & S_{16} & S_{36} \\ S_{16} & S_{17} & S_{20} & S_{24} & S_{28} & S_1 \\ S_{20} & S_{21} & S_{24} & S_{28} & S_{32} & S_5 \\ S_{24} & S_{25} & S_{28} & S_{32} & S_{36} & S_9 \end{bmatrix},$$

Since $\nu = 5$, then $S_0 = 1$ and in this matrix we have that the unknown syndromes are S_5**,** S_{10} **and** S_{20}, but we have

$$S_{10} = S_{2\times 5} = S_5^2 \;\; and \;\; S_{20} = S_{2\times 2\times 5} = S_{4\times 5} = S_5^4$$

Substitute with S_{10} and S_{20} in the matrix and use **Magma or Maple 13** to get a polynomial $f(S_5)$ in single unknown syndrome as follows:

$$f(S_5) = \sum_{i=0}^{11} c_i S_5^i.$$

The leading coefficient of $f(S_5)$ is $c_{11} = S_{36}S_{32} + S_{12}S_9$ always not equal zero which shown by a full computer search, so we have $f(S_5)$ is of degree 11.
Again choose $I_2 = \{0, 1, 2, 4, 8, 12\}$ and $J_2 = \{0, 2, 4, 6, 8, 16\}$ to obtain the matrix

$$S(I, J) = \begin{bmatrix} S_0 & S_1 & S_2 & S_4 & S_8 & S_{12} \\ S_2 & S_3 & S_4 & S_6 & S_{10} & S_{14} \\ S_4 & S_5 & S_6 & S_8 & S_{12} & S_{16} \\ S_6 & S_7 & S_8 & S_{10} & S_{14} & S_{18} \\ S_8 & S_9 & S_{10} & S_{12} & S_{16} & S_{20} \\ S_{16} & S_{17} & S_{18} & S_{20} & S_{24} & S_{28} \end{bmatrix},$$

substitute with S_{10} and S_{20} in the matrix and use **Magma or Maple 13** to get a polynomial $g(S_5)$ in single unknown syndrome as follows:

$$g(S_5) = \sum_{i=0}^{11} d_i S_5^i.$$

The leading coefficient of $g(S_5)$ is $d_{11} = S_8 + S_2S_6$ always not equal zero which shown by a full computer search, so we have $g(S_5)$ is of degree 11.
Note that the coefficient of $f(S_5)$ and $g(S_5)$ are given in [5]

There are ν error if the following holds:

(i) $f(S_r)$ and $g(S_r)$ has the same degree

(ii) $F(S_r) = gcd(f(S_r), g(S_r))$ has the same unknown syndrome S_r as a root, which of degree one.

After this we apply Euclidean Algorithm to find $F(S_r)$, which means that we find S_r and then as in previous cases to have $\sigma_1, \ldots, \sigma_\nu$ For $\nu = 5$ we show that condition (i) hold, if condition (ii) holds, then we can find S_r and then find $\sigma_1, \ldots, \sigma_5$

Chapter 4

Decoding the binary Golay code

The binay Golay code $\mathcal{G}_{23}$ is an important example of a perfect code. It has length 23*, dimension* 12*, and minimum distance* 7*. The binary Golay code $\mathcal{G}_{23}$ is obtained from $\mathcal{G}_{23}$ simply by omitting the last coordinate position from all codewords.In fact, we can omit any one of coordinate positions by the following theorem as in [12] and [14]. There are many known decoding methods for $\mathcal{G}_{23}$ in [12]and [14], For example, being a cyclic code or,a quadratic residue code. In section one we define $\mathcal{G}_{24}$, and then obtain $\mathcal{G}_{23}$. In section two we give a decoding method for the binary Golay code. In section three, we study the factorization of $x^{23}-1$ and some theorem's which important to decode the golay code by using the generator matrix which is studied in final section.*

4.1 Construction of the binary (23, 12, 7) Golay code $\mathcal{G}_{23}$

In this section we will define the $(24, 12, 8)$*code, which is the binary Golay code denoted by $\mathcal{G}_{24}$, and obtain* $(23, 12, 7)$ *Golay code by puncturing $\mathcal{G}_{24}$.*
let $\mathcal{G}_{24}$ be the $[24, 12]$ code with generator matrix $G_{24} = [I_{12}|A]$ in standard form

were I_{12} is the identity matrix and A is a matrix of size 12×12 defined by

$$A = \left(\begin{array}{c|cccccccccc} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

★ Construction of $\mathcal{G}_{24}$

Lable the columns of A by $\infty,\ 0,\ 1, 2, \cdots 10$. The first row contains 0 in column ∞ and 1 elsewhere. To obtain the second row, a 1 is placed in column ∞ and a 1 is placed in columns $0, 1, 3, 4, 5$ and 9, these numbers are the squares of the integers modulo 11. That is $0^2 = 0, 1^2 = 10^2 \equiv 1\ (mod\ 11),\ 2^2 \equiv 9^2 \equiv 4\ (mod\ 11)$ etc.
The first third row of A is obtained by putting a 1 in column ∞ and then shifting the components in the second row one place to the left and wrapping the entry in column 0 around to column 10, and so on all other rows.

Remark 4.1.1. [23]

(1) All rows has weight divisible by 4 and $dim\ \mathcal{G}_{24} = 12$.

(2) $\mathcal{G}_{24}$ is self-dual binary code; i.e $\mathcal{G}_{24} = \mathcal{G}_{24}^{\perp}$

(3) The minimum weight of $\mathcal{G}_{24}$ is 8

★ Puncturing codes

Definition 4.1.1. [23] Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$, we can puncture $\mathcal{C}$ by deleting the same coordinate i in each codeword. The puncture code of $\mathcal{C}$ denoted by $\mathcal{C}^*$ has length $n-1$.

Theorem 4.1.1. *[23]Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$, and let $\mathcal{C}^*$ be the code $\mathcal{C}$ punctured on the ith coordinate.*

(i) If $d > 1$, $\mathcal{C}^*$ is an $[n-1, k, d^*]$ code where $d^* = d-1$ if $\mathcal{C}$ has a minimum weight codeword with a non zero ith coordinate and $d^* = d$ otherwise.

(ii) When $d = 1$, $\mathcal{C}^*$ is an $[n-1, k, 1]$ code, if $\mathcal{C}$ has no codeword of weight 1 whose nonzero entry is in coordinate i; otherwise, if $k > 1$, $\mathcal{C}^*$ is an $[n-1, k-1, d^*]$ code with $d^* \geq 1$.

Example 4.1.1. *[23] Let $\mathcal{C}$ be the $[5, 2, 2]$ binary code with generator matrix*

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

Let $\mathcal{C}_1^*$ and $\mathcal{C}_5^*$ be the code $\mathcal{C}$ punctured on coordinate 1 and 5, respectively, they have generator matrices

$$G_1^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \quad G_5^* = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

So $\mathcal{C}_1^*$ is a $[4, 2, 1]$ code, while $\mathcal{C}_5^*$ is a $[4, 2, 2]$ code.

Example 4.1.2. *[23] Let $\mathcal{D}$ be the $[4, 2, 1]$ binary code with generator matrix*

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

Let $\mathcal{D}_1^*$ and $\mathcal{D}_4^*$ be the code $\mathcal{D}$ punctured on coordinate 1 and 4, respectively, they have generator matrices

$$D_1^* = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \quad D_4^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

So $\mathcal{D}_1^*$ is a $[3, 1, 3]$ code, while $\mathcal{D}_4^*$ is a $[3, 2, 1]$ code.

In general a code $\mathcal{C}$ can be punctured on the coordinate set T by deleting T components in all codewords of $\mathcal{C}$. If T has size t, the resulting code, which we will often denote $\mathcal{C}^T$ is an $[n-t, k^*, d^*]$ where $k - t \geq k^* \geq k,\ d - t \geq d^* \geq d$

★ **Construction of $\mathcal{G}_{23}$**

If we puncture in any of the coordinates of $\mathcal{G}_{24}$ we obtain a $[23, 12, 7]$ binary code denoted by $\mathcal{G}_{23}$ called binary Golay code has minimum weight 7. **Note that :** The extended code of $\mathcal{G}_{23}$ is $\mathcal{G}_{24}$ so $\mathcal{G}_{24}$ is called extended Golay code.

4.2 Decoding the (23, 12, 7) binary Golay code

There are more than one way to decode the binary Golay code, as example a decoding method described in [14]. In this section we study a different decoding method for $(23, 12, 7)$ *binary Golay code.*

Let $n = p = 23$ be odd prime where $n = p = 23 \equiv -1 \bmod 23$. As described in Section 3.2 and Section 3.3 by **Maple 13** we have the following:

(1) The set of quadratic residue modulo 23, is

$$Q_{23} = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}.$$

(2) The smallest extension field of $\mathbb{F}_2$ is $\mathbb{F}_{2^{11}} = GF(2^{11})$.

(3) $f(x) = x^{11} + x^2 + 1$ is irreducible polynomial over $\mathbb{F}_2$.

(4) $\beta = \alpha^{(2^m-1)/n} = \alpha^{(2^{11}-1)/23} = \alpha^{89}$ be a primitive 23th root of unity in $GF(2^{11})$.

(5) The generator polynomial of binary (23, 12, 7) Golay code is

$$g(x) = \prod_{i \in Q_{23}} (x - \beta^i) = \prod_{i \in Q_{23}} (x - \alpha^{89^i}) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1.$$

The base set of the (23, 12, 7) QR code is $S = \{1, 5\}$, where

$$C_1 = \{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$$

$$C_5 = \{5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14\}$$

Since $t = \frac{(d-1)}{2} = 3$, this code allows for the correction of up to three errors. i.e we have four cases

Case 0: No error in the received codeword if and only if $S_1 = 0$; otherwise go to case 1.

Case 1: One error in the received codeword if and only if the two conditions in Theorem 3.1.2 holds, i.e. $S_5 = S_1^5$ and $S_1^{23} = 1$; otherwise go to case 2.

Case 2: Two errors in the received codeword if and only if the two conditions in Theorem 3.1.3 holds. i.e want to find $S_{-1} = S_{22}$ for this choose two sets $I_2 = \{0, 4, 6\}$ and $J_3 = \{0, 2, 18\}$, yields the matrix $S(I_2, J_2)$ of size 3×3 as follows

$$\begin{bmatrix} S_0 & S_2 & S_{18} \\ S_4 & S_6 & S_{22} \\ S_6 & S_8 & S_1 \end{bmatrix}$$

We have $det(I_2, J_2) = 0$, so

$$S_4(S_1S_2 + S_8S_{18}) + S_6(S_0S_1 + S_6S_{18}) + S_{22}(S_0S_8 + S_2S_6) = 0$$

substitute $S_2 = S_1^2$, $S_4 = S_1^4$, $S_8 = S_1^8$, $S_6 = S_3^2$, $S_{18} = S_9^2$ to get

$$S_1^4(S_1^3 + S_1^8S_9^2) + S_3^2(S_3^2S_9^2) + S_{22}(S_1^2S_3^2) = 0$$

$$S_1^7 + S_1^{12}S_9^2 + S_3^4S_9^2 + S_{22}S_1^2S_3^2 = 0$$

$$S_{22} = \frac{S_1^7 + S_1^{12}S_9^2 + S_3^4S_9^2}{S_1^2S_3^2}.$$

Now you can apply the two conditions in Theorem 3.1.3 holds, if they dont holds, go to case 3.

Case 3: For three errors, From Newton identities we have

$$S_1 + \sigma_1 = 0 \text{.....(1)}$$

$$S_3 + \sigma_1 S_1^2 + \sigma_2 S_1 + \sigma_3 = 0 \text{.....(2)}$$

$$S_5 + \sigma_1 S_1^4 + \sigma_2 S_3 + \sigma_3 S_1^2 = 0 \text{.....(3)}$$

$$S_7 + \sigma_1 S_3^2 + \sigma_2 S_5 + \sigma_3 S_1^4 = 0 \text{.....(4)}$$

$$S_9 + \sigma_1 S_1^8 + \sigma_2 S_7 + \sigma_3 S_3^2 = 0 \text{.....(5)}$$

$$S_9 + S_1^9 + \sigma_2(\sigma_1 S_3^2 + \sigma_2 S_5 + \sigma_3 S_1^4) + \sigma_3 S_3^2 = 0$$

$$S_9 + S_1^9 + \sigma_2\sigma_1 S_3^2 + \sigma_2^2 S_5 + \sigma_2\sigma_3 S_1^4) + \sigma_3 S_3^2 = 0$$

Substitute $S_5 = \sigma_1 S_1^4 + \sigma_2 S_3 + \sigma_3 S_1^2$ to get

$$S_9 + S_1^9 + \sigma_2\sigma_1 S_3^2 + \sigma_2^2(\sigma_1 S_1^4 + \sigma_2 S_3 + \sigma_3 S_1^2) + \sigma_2\sigma_3 S_1^4 + \sigma_3 S_3^2 = 0$$

$$S_9 + S_1^9 + \sigma_2\sigma_1 S_3^2 + \sigma_2^2 S_1^5 + \sigma_2^3 S_3 + \sigma_2^2\sigma_3 S_1^2 + \sigma_2\sigma_3 S_1^4 + \sigma_3 S_3^2 = 0$$

$$S_9 + S_1^9 + \sigma_2\sigma_1 S_3^2 + \sigma_2^2 S_1^5 + \sigma_2^3 S_3 + \sigma_3(\sigma_2^2 S_1^2 + \sigma_2 S_1^4 + S_3^2) = 0$$

Substitute $\sigma_3 = (S_3 + S_1^3) + \sigma_2 S_1$ to get

$$S_9+S_1^9+\sigma_2\sigma_1 S_3^2+\sigma_2^2 S_1^5+\sigma_2^3 S_3+(S_3+S_1^3)(\sigma_2^2 S_1^2+\sigma_2 S_1^4+S_3^2)+\sigma_2^3 S_1^3+\sigma_2^5 S_1^5+\sigma_2 S_1 S_3^2 = 0$$

$$S_9 + S_1^9 + (S_3 + S_1^3)(\sigma_2^3 + \sigma_2^2 S_1^2 + \sigma_2 S_1^4 + S_3^2) = 0$$

$$S_9 + S_1^9 = (S_3 + S_1^3)(\sigma_2^3 + \sigma_2^2 S_1^2 + \sigma_2 S_1^4 + S_3^2)$$

$$S_9 + S_1^9 = (S_3 + S_1^3)[(\sigma_2 + S_1^2)^3 + S_1^6 + S_3^2]$$

So

$$\sigma_2 = S_1^2 + D^{\frac{1}{3}} \quad and \quad \sigma_3 = S_3 + S_1 D^{\frac{1}{3}}$$

Where $D = S_1^6 + S_3^2 + \frac{(S_9+S_1^9)}{(S_3+S_1^3)}$

4.3 The group of a code

In this section, we give some theorems which is important to decode the binary Golay code with generator polynomial.

Theorem 4.3.1. *[12] A binary* $(23, 12, 7)$ *code is unique (up to equivalence).*

$\mathcal{G}_{23}$ can be constructed in a more natural way as a cyclic code as follows. Let $\mathcal{R} = \mathbb{F}_2[x]/(x^{23} - 1)$. The factorization of $x^{23} - 1$ into irreducible polynomials in $\mathbb{F}_2[x]$ is given by

$$x^{23} - 1 = (x - 1)g_1(x)g_2(x)$$

with

$$g_1(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$
$$g_2(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

The cyclic code $\mathcal{C}_1 = \langle g_1(x) \rangle \subset \mathcal{R}$ and $\mathcal{C}_2 = \langle g_2(x) \rangle \subset \mathcal{R}$ can be shown to be all equivalent to $\mathcal{G}_{23}$. The idempotent generator for $\mathcal{C}_1$ and $\mathcal{C}_2$can be taken to be

$$n(x) = x^5 + x^7 + x^{10} + x^{11} + x^{14} + x^{15} + x^{17} + x^{19} + x^{20} + x^{21}$$
$$q(x) = x + x^2 + x^3 + x^4 + x^6 + x^8 + x^9 + x^{12} + x^{13} + x^{16} + x^{18}$$

Since the order of 2 modulo 23 is 12, the quadratic residues Q and the non residues N modulo 23 are

$$Q = \langle 2 \rangle = \{1, 2, 4, 8, 16, 9, 18, 3, 6, 12\},$$

$$N = 5\langle 2 \rangle = \{5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14\}.$$

Note that the exponents which appear in $q(x)$ are exactly the quadratic residues and those in $n(x)$ are quadratic non residues. Thus $\mathcal{G}_{23}$ is also a **quadratic residue code**.

The group of a code $\mathcal{C}$ is useful in determining the structure of the code, computing weight distributions, classifying codes, and devising decoding algorithms.

If $\mathbf{v}= (v_1, \ldots, v_n)$ is a vector and ϕ is a permutation on n objects, then ϕ send $\mathbf{v}$ into $\mathbf{v}\phi=\mathbf{w}= (w_1, \ldots, w_n)$ with $v_i = w_{i\phi}$. Every permutation of the n coordinate positions sends $\mathcal{C}$ onto an equivalent [n, k]-code or onto itself. It is easy to check that the set of all permutations that send $\mathcal{C}$ onto itself is a group. This group is called the group of $\mathcal{C}$. It is denoted by $G(\mathcal{C})$.

Clearly any element in $G(\mathcal{C})$ applied to the coordinate positions of any generator matrix of $\mathcal{C}$ yields another generator matrix of $\mathcal{C}$. The group of $\mathcal{C}$ is a subgroup of S_n.

We can now say that a length n code $\mathcal{C}$ is cyclic if the group of $\mathcal{C}$ contains the cyclic group of order n generated by $\sigma = (0, 1, \ldots, n-1)$.

However, $G(\mathcal{C})$ might be, and usually is, larger than this as we see from the following theorem

Theorem 4.3.2. *[21] Let $\mathcal{C}$ be an odd length n binary cyclic code. Let $\sigma \in S_n$ be the cyclic shift, that is,$(i)\sigma = (i + 1)$ (mod n) and $\tau \in S_n$ be the permutation defined by $(i)\tau = 2i$ (mod n). Both σ and τ are considered to act on $0, 1, \ldots, n-1$. Let m be the order 2 mod n. Then $\tau\sigma\tau^{-1} = \sigma^{2^{m-1}}$ and $\tau^{-1}\sigma^i\tau = \sigma^{2i}$ for $0 \leq i \leq n-1$. Furthermore, τ is in $G(\mathcal{C})$, and hence the group P generated by σ and τ is a subgroup of $G(\mathcal{C})$. The order of P is mn.*

Definition 4.3.1. [21] If G is a generator matrix of an [n, k]-code $\mathcal{C}$, then any set of k columns of G that are independent is called an information set of $\mathcal{C}$.

Note that any permutation π in $G(\mathcal{C})$ sends an information set into an information set. We may take the information set for $\mathcal{G}_{23}$ to be $\{11, 12, \ldots, 22\}$ for an appropriate generator matrix.

Theorem 4.3.3. *[21] Let $\sigma : i \to i+1 \ (mod \ \ 23)$, and $\tau : i \to 2i \ (mod \ \ 23)$. Then $P = \langle \sigma, \tau \rangle$ is a subgroup of $\mathcal{G}_{23}$ such that for any error vector e of weight≤ 3, some $\pi_i \in P$ moves all the $1's$ in e out of the information places.*

Proof. Let $e = e_0 e_1 \dots e_{22}$ be an error vector of weight ≤ 3. We need to show that some $\pi \in P$ moves all the 1's in e out of the information places.
Applying cyclic shift σ, we may assume that $E = i|e_i = 1 = \{0, l, k\}$, without loss of generality. As before, the quadratic residues Q and nonresidues N modulo 23 are

$$Q = \langle 2 \rangle = \{1, 2, 4, 8, 16, 9, 18, 3, 6, 12\},$$

$$N = 5\langle 2 \rangle = \{5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14\}$$

Therefore, if $l \in Q$, then there is some i such that $2^i l = 1$ and if $l \in N$, then there is some i such that $2^i l = 5$. Thus by applying τ^i, we may assume that $E = \{0, 1, k\}$ or $E = \{0, 5, k\}$. Since $\sigma \in P$, it suffices to show that there is i such that $E\tau^i = \{0, a, b\}$ $a < b$ satisfying $a > 11$ or $b - a > 11$ or $22 - b > 11$.

1. Suppose $E = \{0, 1, k\}$. If $k < 10$ or $k > 13$, then we are done. If $k = 10$ or $k = 11$, then apply τ to E to get $E_\tau = \{0, 2, 22\}$ or $\{0, 2, 1\}$
2. Suppose $E = \{0, 5, k\}$. If $k < 10$ or $k > 17$, then we are done, again. For other cases, one more application of τ is enough as we can see in the table below

E	E_τ
$\{0, 5, 11\}$	$\{0, 10, 22\}$
$\{0, 5, 12\}$	$\{0, 1, 10\}$
$\{0, 5, 13\}$	$\{0, 3, 10\}$

$\{0, 5, 14\}$ $\{0, 5, 10\}$

$\{0, 5, 15\}$ $\{0, 7, 10\}$

$\{0, 5, 16\}$ $\{0, 9, 19\}$

□

4.4 A decoding method for $\mathcal{G}_{23}$ using the generator matrix

In this section, we refer to Theorem 2.1.4 *to use the generator matrix to decode the binary Golay code*

Suppose a codeword $\mathbf{x} = x_0x_1 \dots x_{22}$ is transmitted, an error vector $\mathbf{e} = e_0e_1 \dots e_{22}$ occur with weight ≤ 3, and the vector $\mathbf{y}$=$\mathbf{x}$+$\mathbf{e}$= $y_0y_1 \dots y_{22}$ is received. Let G be the generator matrix of $\mathcal{G}_{23}$ such that $\{11, 12, \dots, 22\}$ is an information set. Hence x_L= $x_0x_1 \dots x_{10}$ are the **check symbols**, and x_R= $x_{11}x_{12} \dots x_{22}$ are the **information symbols**. Write $G = (G_L|G_R)$, where G_L is a 12×11-matrix and G_R is a 12×12-matrix.Then G_R is invertible.

Now there exists some $\pi_i \in P$ such that $\mathbf{y}_i$=$\mathbf{y}\pi_i$ has no errors in the information places. Since $(y_i)_R$ is the information symbols, there exists a unique codeword $\mathbf{w}$ such that $\mathbf{w}_R$=$\mathbf{y}_iR$. In fact $\mathbf{w} = \mathbf{x}\pi_i$ since $\mathbf{x}\pi_i \in \mathcal{G}_{23}$ and d($\mathbf{x}\pi_i$, $\mathbf{y}\pi_i$)=d($\mathbf{x}$, $\mathbf{y}$)≤ 3. Recall that the encoding map $\mathbf{u}\mapsto \mathbf{uG}$ from $\mathbb{F}_2^{12}$ to $\mathcal{G}_{23}$ is bijective. Thus there exists a unique vector $\mathbf{u} \in \mathbb{F}_2^{12}$ such that $\mathbf{uG}$=$\mathbf{w}$, and then we have
($\mathbf{uG}_L|\mathbf{uG}_R$)=$\mathbf{uG}$=$\mathbf{w}$=($\mathbf{w}_L|\mathbf{w}_R$)=($\mathbf{w}_L|$($\mathrm{y}_i$)$_R$).

Hence $\mathbf{u}$=$(\mathbf{y})_{iR}(\mathrm{G}_R)^{-1}$ and $\mathbf{w}_L$=$\mathbf{uG}_L$=$(\mathrm{y}_i)_R(\mathrm{G}_R)^{-1}\mathbf{G}_L$. Consequently,
$\mathbf{x}$=$\mathbf{w}{\pi_i}^{-1}$=$((\mathrm{y}_i)_R(\mathrm{G}_R)^{-1}\mathbf{G}_L|(\mathrm{y}_i)_R){\pi_i}^{-1}$.

★ **The decoding procedure**

When $\mathbf{y}$ is received, each $\mathbf{y}_i$=$\mathbf{y}\pi_i$ and

$\mathbf{w}_L$=$\mathbf{y}_i(\mathrm{G}_R)^{-1}\mathbf{G}_L$

in turn is computed, until an i is found for which $\mathrm{d}(\mathbf{w}_L, (\mathbf{y}_i)_L) \leq 3$ Then the errors are all in the first 11 places of $\mathbf{y}\pi_i$, and we decode $\mathbf{y}$ as

$\mathbf{x}$=$(\mathbf{w}_L|(\mathbf{y}_i)_R)\pi_i{}^{-1}$

If $\mathrm{d}((\mathbf{w}_L - \mathbf{y}_i)_L) > 3$ for all i, we conclude that more than 3 errors have occurred.

Example 4.4.1. *[21] In Theorem* 4.3.1*, we found that*

$$g_1(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$$

This polynomial determines the generator matrix $\boldsymbol{G} = [\boldsymbol{G}_L|\boldsymbol{G}_R]$ *for* $\mathcal{G}_{23}$ *with*

$$G_L = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$G_R = \left[\begin{array}{cccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}\right].$$

and so $\boldsymbol{D}=\boldsymbol{G_R}^{-1}\boldsymbol{G_L}$

$$D = \left[\begin{array}{cccccccccccc}
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}\right].$$

Suppose that $\boldsymbol{x}$= (01000010110|111100000000) *was sent and the vector*

$\boldsymbol{y}$= (01010010110|111001000000) *is received. Since* $\boldsymbol{y}_R D$= 11100011011 *and d(*$\boldsymbol{y}_L$,$\boldsymbol{y}_R D$*)*= 6, $\boldsymbol{y}$ *has some errors. We compute* $\boldsymbol{y}\pi^i$ *for* $\pi^i \in P = \langle \sigma, \tau \rangle$ *and* $\boldsymbol{w}_L$=$(\boldsymbol{y}_i)_R D$ *until an* i *is found for which d(*$\boldsymbol{w}_L$, $(\boldsymbol{y}_i)_L$*)*≤ 3. *The existence of such* π_i *is guaranteed by Theorem* 4.3.3. *Note that* $E = \{3, 14, 16\}$ *and*

$$E\sigma^9\tau = \{12, 0, 2\}\tau = \{1, 0, 4\}$$

At some stage, with $\pi_i = \sigma^9\tau$, *we will compute* $\boldsymbol{y}_i$=$\boldsymbol{y}\pi^i$=$\boldsymbol{y}\sigma^9\tau$= (01001001000|101000101110) *and* $\boldsymbol{w}_L$=$(\boldsymbol{y}_i)D$= 10000001000 *and find that d(*$\boldsymbol{w}_L$, $(\boldsymbol{y}_i)_L$*)*= 3. *Thus we decode* $\boldsymbol{y}$ *as* $\boldsymbol{x}$=$(\boldsymbol{w}_L$, $(\boldsymbol{y}_i)_R))\pi_i^{-1}$=(10000001000|101000101110)$\tau^{-1}\sigma^{-9}$

$$= (10000000001|0000101101011)\sigma^{-9}$$

$$= (01000010110|111100000000).$$

Conclusion

In this thesis, we have studied a new algebraic decoding for $(41, 21, 9)$ **quadratic residue code of odd length in the form** $n = 8l + 1$**, where l is integer. Some new more general properties are found for the syndromes of the subclass of binary QR codes, where every known syndromes (respectively , unknown syndromes) can be expressed as some power of** S_1 **(respectively.,**S_3**).**

A new technique is found to have the unknown syndrome S3 which is a necessary condition for decoding the (41, 21, 9) QR code.

Index

Zero-divisor 6

zero code 13

Bibliography

[1] Chang Y., Chen Y. H., He R., Lee C. D., Reed I. S. and Truong T. K., "*Algebraic Decoding of Quadratic Residue Codes Using Inverse-Free Berlekamp-Massey Algorithm*", Journal of Information Science and Engineering, vol. 23, Jan. 2007, pp. 127-145.

[2] Chang Y., Cheng H. Y.,Lee C.D., Reed I.S. and Truong T.K. : "*Algebraic Decoding of* $(71, 36, 11)$*,* $(79, 40, 15)$*, and* $(97, 49, 15)$ *Quadratic Residue Codes*", IEEE Trans. Commun., vol.51, no.9, 2003, pp. 1463-1473 .

[3] Chang Y., Chen Y. H., Lee C. D. and Truong T. K., "*Algebraic Decoding of* $(103, 52, 19)$ *and* $(113, 57, 15)$ *Quadratic Residue Codes*", IEEE Trans. Commun., vol. 53, May 2005, pp. 749-754.

[4] Chen X., He R., Reed I.S. and Truong T.K. : "*Decoding the* $(47, 24, 11)$ *Quadratic Residue Code*", IEEE Trans. Inf. Theory, vol.47 , no.3, 2001, pp. 1181 -1186.

[5] Chen X., Reed I.S. and Truong T.K. : :*Decoding the* $(73, 37, 13)$ *Quadratic Residue Code*:, IEE Proc. Comput. Digit. Tech. , vol.141, no.5,1994, pp. 974-986.

[6] Chen X., Reed I.S., Truong T.K. and Yin X. : "*The Algebraic Decoding of the* $(41, 21, 9)$ *Quadratic Residue Code*", IEEE Trans. Inf. Theory,vol.38 , no.3, 1992, pp. 974-986.

[7] D.G.Hoffman, D.A.Leonard, C.C.Lindner, K.T.p helps , C.A.Rodger and J.R.Wall "*Coding theory The Essentials*", Printed in the United State Of America 1991.

[8] D. Hofman, " *Coding theory*", Markel Dekker, 1990.

[9] I.N.Herstein, "*Topics in Algebra*",University of Chicago.press, 1975.

[10] J.H.Conway and M.A.Odlyzko, "*Sphere Packings*", Springer-Verlag, 1993.

[11] Joseph A. Gallian "*Contemporary abstract algebra*",Oxford University Press,2^{nd} ed., 1992.

[12] J.Y Yoon, Y. H Park, "*A decoding method for the binary golay code*", Kangweon-Kyungki Math. Jour. no.1, 2002, pp. 89-95 .

[13] Lin T.C, Shih P.Y., Su W.K. and Truong T.K. : "*Decoding of the* $(41, 21, 9)$ *Quadratic Residue Code using the Gaos Algorithm*", IEEE Trans. Inf. Theory, vol.38, Issue 3, 2008, pp. 974-985 .

[14] Lin T. C, Shih P. Y., Su W. K., and Truong T. K., "*Soft-decoding of the* $(23, 12, 7)$ *binary Golay code*", IEEE Trans. Commun., vol. 2, March 2008.

[15] M. J. E. Golay, "*Notes on digital coding*", Proc. IRE, vol. 37, 1949, pp. 67

[16] N.J.A.Sloane and J.H.Conway, "*Soft decoding techniques for codes and lattices*", including the Golay code and the Leech lattice, PGIT 32, 1986, pp. 41-50.

[17] Prange E., Reed I.S. and Truong T.K. : "*Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms*", Air Force Cambridge Research Center, Cambridge, 1958, pp.58-156

[18] R. W. Hamming, "*Error detecting and error correcting codes*", Bell Syst. Tech. J., vol. 29, 1950, pp. 147-160.

[19] San ling and Chaoping xing "*Coding Theory A first Course*", Cambridge University press, 2004.

[20] V.Pless, "*Decoding the Golay codes*", PGLT32, 1986, pp. 561-567.

[21] V.Pless, "*Introduction to the Theory of Error-Correcting Codes*", A Wiley- Interscience Publication, 1989.

[22] V.Pless, "*On the uniqueness of the Golay codes*", JCT5, 1968, pp. 215-228.

[23] W.C. Huffman and V. Pless, " *Fundementals of Error-Correcting Codes*", Cambridge, U.K.:Cambridge, 2003.