BACHELOR'S DEGREE THESIS

Degree in Mathematics

Title: Maximum Distance Separable Codes

Author: Albanell Sarroca, Elisabeth

Advisor: Ball, Simeon

Department: Matemàtica Aplicada IV

Academic year: 2013/2014

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Bachelor Thesis

Maximum Distance Separable Codes

Elisabeth Albanell Sarroca

Advisor: Simeon Ball

Matematica Aplicada IV

A tutti coloro che mi hanno sempre sostenuto, nei momenti buoni, ma anche in quelli cattivi. Senza di loro nulla sarebbe stato possibile.

Abstract

Keywords: Linear codes, Singleton bound, p-adic linear codes, MDS codes

MSC2000: 94B05, 94B65, 94B60, 51E22

The focus of this bachelor thesis are maximum distance separable codes. A code is used to communicate over noise channel so any interferences that may occur can be detected and corrected. Maximum distance separable codes have a capacity to correct many errors. Maximum distances separable codes are usually constructed as linear codes over fields but in this text we will also consider them over certain commutative rings. Recently it has been proven that all linear maximum distance separable codes over prime fields are short and we will prove that this carries over to certain linear maximum distance separable codes over p-adic rings.

Acknowledgments

I would like to take the opportunity to express my gratitude to my supervisor, Simeon Ball, whose understanding, patience and willingness to help, added to his undeniable mastery and years of experience in working on MDS codes, have made writing this dissertation possible. I appreciate his vast knowledge and skills, and his assistance during all these months. Even when I was living Rome he was always available.

Besides my advisor, I would like to thank the whole Naranjo Barnet family, especially Joan Carles who has also read this dissertation and gave advice. They have all been of great help not only for writing this dissertation but also during the whole degree.

A very special thanks goes out to my FME fellows who have been more than friends during all these years and have become more like family. Without their motivation and encouragement I would not have made it. I would like to name, Marc Calvo, Jordi Pizarro, Pol Naranjo, Pere Daniel Prieto and Ruben León in particular.

Come la maggior parte del tempo per fare questa tesi ho vissuto a Roma, non vorrei perdere l'occasione per ringraziare per tutto lo aiuto, sostegno e tutti i bei momenti le mie Roma ragazze, Denise, Pinar, Anche e Francesca, ed anche a tutti gli amici che ho fatto là, soprattutto Abel.

Last but not least, I would like to thank all my family, especially my mother Lupe, my father Tete and my sister Marta as well as my friends Judith, Marta P., Silvia, Belen and Marta F. They may be not have supported me in a mathematical way but they have given me support through my entire life and have encouraged me to never give up on my goals.

Contents

Chapter 1
Basic concepts and definitions

A basic outline of a general communication process is illustrated in Figure 1.1.

Error Correcting Codes Theory deals with the second and forth step of that outline, which are the coding and decoding processes and focuses on the problem of detecting and correcting errors in the received message. *Coding theory* should not be confused with Cryptography. While in the latter the aim is to send a message to a friendly receiver in the safest way, in the former it is to send it in the most efficient way while being able to correct errors caused by noise. In this way, the main aim of Error Correcting Codes theory is to construct "good codes". A code is considered to be a good code when it enables us to codify many messages (big size), that can be sent in a fast and efficient way (which means that they have a high **code rate** as will be defined later), when it detects and corrects at the same time the largest number of errors as possible (which means that have the biggest possible **minimum distance** δ as will be defined later) and for which there exists easy and effective decoding algorithms. Those aims are hardly ever possible to satisfy at the same time since they are contradictory, so all in all the aim is to find a balance between all the parameters involved or given some of the parameters, find the other one that makes the code as good as possible.

DEFINITION 1.1. A *q-ary alphabet* $A = \{a_1, ..., a_q\}$ is a finite non empty set of cardinality q. The elements of A are called *letters* or *symbols*. For each $n \geq 1$, the elements of A^n are written as $(a_1, a_2, ..., a_n)$ or $a_1a_2...a_n$ indistinctly and are called

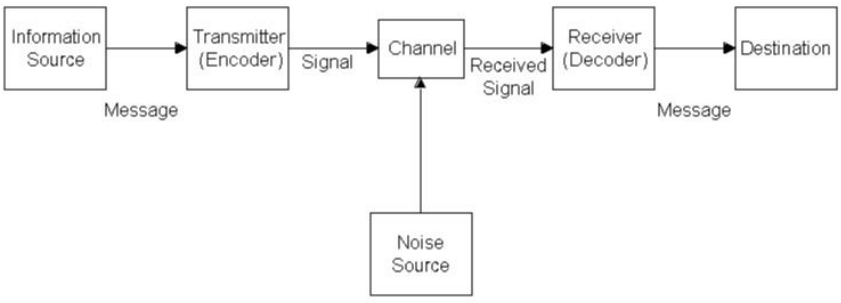

FIG. 1.1. Schematic diagram of a general communication system.

words of *length* n in the alphabet A. In this text A^0 will have exactly one word, that will be called λ, which is called *empty word* and which length is 0. It will be represented as A^* the set of all the words of an alphabet A, that is,

$$A^* = \bigcup_{n \geq 0} A^n.$$

DEFINITION 1.2. Let $A = \{a_1, ..., a_q\}$ be an alphabet, a q-ary *code* over A is a subset C of A^*. The elements of $\mathcal{C}$ are called *codewords*.

In this text we will try to use $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ for general words and $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ for codewords. The number $M = |\mathcal{C}|$ is called the *size* of the code.

When $q = 2$, $q = 3$, $q = 4$ they are called *binary*, *ternary* and *quaternary codes* respectively.

The words in a code can have variable length, but if they all have the same length n it is called a *block code of length* n and $\mathcal{C}$ can be said to be a $(n, M) - code$.

DEFINITION 1.3. The *code rate* of a q-ary $(n, M) - code$ is

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

The code rate gives an idea of the proportion of the data-stream that is useful (non-redundant), in other words the percentage of digits that contain information from the original message out of the ones that have been sent. If the code rate is $\frac{k}{n}$, for every k bits of useful information, the code generates n bits of data in total, of which $n - k$ are redundant. Therefore, it is then common to try to construct codes that have a high code rate ($R > \frac{2}{3}$ or $R > \frac{3}{4}$).

DEFINITION 1.4. Given two words $\mathbf{x}$ and $\mathbf{y}$ of an alphabet A^n, the distance between them $d(\mathbf{x}, \mathbf{y})$ is defined to be the number of positions in which $\mathbf{x}$ and $\mathbf{y}$ differ, that is if $\mathbf{x} = x_1...x_n$ and $\mathbf{y} = y_1...y_n$, then $d(\mathbf{x}, \mathbf{y})$ is the number of values i for which $x_i \neq y_i$. This distance function is called the *Hamming distance*,

$$d(\mathbf{x}, \mathbf{y}) = \#\{i \in [n] : x_i \neq y_i\}$$

and as a distance it satisfies the following properties:

- $d(\mathbf{x}, \mathbf{y}) \geq 0$
- $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$
- $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
- The triangle inequality ($d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$)

DEFINITION 1.5. A code $\mathcal{C}$ is said to be *t-error-detecting* if $d(\mathbf{u}, \mathbf{v}) > t$ for any two distinct codewords.

This means that whenever a codeword is changed in at most t of its symbols no other codeword is reached. So if that word is received, it will be posible to tell that it is not a codeword and that some interference has occured in transmission.

DEFINITION 1.6. A code $\mathcal{C}$ is said to be *t-error-correcting* if there do not exist words $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ and $\mathbf{x} \in A^n$ such that $d(\mathbf{u}, \mathbf{x}) \leq t$ and $d(\mathbf{v}, \mathbf{x}) \leq t$.

This means that whenever a codeword is taken and changed in at most t of its symbols no other codeword is reached, and moreover that the word that has been sent could not have been obtained from a different starting codeword by changing at most t of the symbols. In this way the receiver is able to tell which codeword was originally sent.

DEFINITION 1.7. The *distance between an element* $\mathbf{x} \in A^n$ *and a code* $\mathcal{C} \subseteq A^n$, $d(\mathbf{x}, \mathcal{C})$, is defined as the minimum distance between $\mathbf{x}$ and each element of the code,

$$d(\mathbf{x}, \mathcal{C}) = \min\{d(\mathbf{x}, \mathbf{u}) \mid \mathbf{u} \in \mathcal{C}\}.$$

DEFINITION 1.8. The *minimum distance of a code* $\mathcal{C}$, $\delta(\mathcal{C})$, is defined as the minimum distance in the code which is the smallest distance between any pair of distinct codewords,

$$\delta(\mathcal{C}) = \min\{d(\mathbf{u}, \mathbf{v}) \mid \mathbf{u}, \mathbf{v} \in \mathcal{C}, \mathbf{u} \neq \mathbf{v}\}.$$

LEMMA 1.9. *A code* $\mathcal{C}$ *is* t*-error-detecting if and only if* $\delta(\mathcal{C}) > t$*, and is* t*-error-correcting if, and only if,* $\delta(\mathcal{C}) > 2t$*.*

The following, are some of the most important parameters of a code:

- Its length (n).
- Its size (M).
- Its minimum distance (δ).
- The number of symbols of the alphabet ($|A| = q$).

Once all of them are known one can talk about a $(n, M, \delta)_q$*-code.*

EXAMPLE 1.1. The *repetition code* $R_q(n)$ of length n over a q-ary alphabet A is

$$R_q(n) = \{(a, ..., a) \text{ of length } n \mid a \in A\}$$

Hence it is a $(n, q, n)_q$-code. □

EXAMPLE 1.2. Let $\mathcal{C}$ be a $(n, M, \delta)_2$-binary-code. Then from it a new code $\overline{\mathcal{C}}$ formed by the words $(u_1, ..., u_n, u_{n+1})$ such that $(u_1, ..., u_n) \in \mathcal{C}$ and u_{n+1} is equal to 0 or 1 depending on whether the number of ones is even or odd can be constructed. The digit u_{n+1} is called *parity bit*. Therefore the parameters of the new code $\overline{\mathcal{C}}$ are:

- length: $n + 1$.
- size: M
- minimum distance: δ or $\delta + 1$
- number of symbols of the alphabet ($|A| = 2$).

Observe that two codewords $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ are at an even distance if and only if their corresponding codewords in $\overline{\mathcal{C}}$ have the same parity bit. Thus,

- if two codewords $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ are at an even distance δ then their corresponding codewords $\overline{\mathbf{u}}, \overline{\mathbf{v}} \in \overline{\mathcal{C}}$ are at the same distance δ.
- if two codewords $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ are at an odd distance δ then their corresponding codewords $\overline{\mathbf{u}}, \overline{\mathbf{v}} \in \overline{\mathcal{C}}$ are at an even distance $\delta + 1$.

Therefore the distance between any given two words in $\overline{\mathcal{C}}$ is always even and $\overline{\mathcal{C}}$ is either a $(n+1, M, \delta)_2$-code if δ is even or a $(n+1, M, \delta+1)_2$-code if δ is odd. □

Another important concept that will be very useful when talking about linear codes is the *weight of a word*.

DEFINITION 1.10. Given a code $\mathcal{C}$ and a codeword $\mathbf{u}$, the *weight* of the word is defined to be the number of non-zero symbols in $\mathbf{u}$,

$$w(\mathbf{u}) = \#\{i \in [n] \mid u_i \neq 0\},$$

or equivalently $w(\mathbf{u}) = d(\mathbf{u}, \mathbf{0})$. In particular, the weight of the word $\mathbf{0}$ is 0.

Therefore it is obvious that the distance between two codewords is equal to the weight of one minus the other $d(\mathbf{u}, \mathbf{v}) = w(\mathbf{u} - \mathbf{v})$ since a coordinate in two different codewords $\mathbf{u}$ and $\mathbf{v}$ are equal, if and only if, the corresponding coordinate of the word $\mathbf{u} - \mathbf{v}$ is 0.

DEFINITION 1.11. The *minimum weight of a code* can be defined as

$$w(\mathcal{C}) = \min\{w(\mathbf{u}) \mid \mathbf{u} \in \mathcal{C}, \mathbf{u} \neq 0\}$$

DEFINITION 1.12. Given $\mathbf{x} \in A^n$, with $|A| = q$ and an integer $r \geq 0$, the *sphere of radius r centered at* $\mathbf{x}$ is defined as

$$S_q(\mathbf{x}, r) = \{\mathbf{y} \in A^n \mid d(\mathbf{x}, \mathbf{y}) = r\}$$

and the *ball of radius r centered at* $\mathbf{x}$ as

$$B_q(\mathbf{x}, r) = \{\mathbf{y} \in A^n \mid d(\mathbf{x}, \mathbf{y}) \leq r\} = \bigcup_{i=0}^{r} S_q(\mathbf{x}, i)$$

Given $\mathbf{x} \in A^n$, there are no words in A^n whose distance to $\mathbf{x}$ is greater than n. For a t such that $0 \leq t \leq n$ the number of words from A^n that are at distance t from $\mathbf{x}$, is the number of ways of choosing the t coordinates in which the word will differ from $\mathbf{x}$, this is $\binom{n}{t}$, multiplied t times by the number of possible diferent values each coordinate can have.$\binom{n}{t}(q-1)^t$. Hence,

$$|B_q(\mathbf{x}, r)| = \binom{n}{0}(q-1)^0 + \cdots + \binom{n}{r}(q-1)^r = \sum_{i=0}^{r} \binom{n}{i}(q-1)^i$$

DEFINITION 1.13. The *packing radius* of a code $\mathcal{C}$, $\rho(\mathcal{C})$, is the largest integer value of $r \geq 0$ such that the set of balls of radius r centered at each codeword of $\mathcal{C}$ are pairwise disjoint

$$\rho(\mathcal{C}) = \max\{r \mid B(\mathbf{u}, r) \cap B(\mathbf{v}, r) = \varnothing, \forall \mathbf{u}, \mathbf{v} \in \mathcal{C}, \mathbf{u} \neq \mathbf{v}\}.$$

DEFINITION 1.14. The *covering radius* of a code $\mathcal{C}$, $\tau(\mathcal{C})$, is the smallest integer value of $r \geq 0$ such that the set of balls of radius r centered at each codeword of $\mathcal{C}$ cover all A^n

$$\tau(\mathcal{C}) = \min\{r \mid \bigcup_{\mathbf{u} \in \mathcal{C}} B(\mathbf{u}, r) = A^n\}.$$

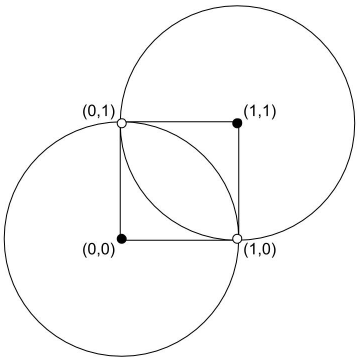

FIG. 1.2. $(2,2,2)_2$-code.

EXAMPLE 1.3. Consider the $R_2(2)$ code which is an $(2,2,2)_2$-code formed by $\mathcal{C} = \{(0,0),(1,1)\}$ illustrated in Figure 1.2 as the coloured dots of the four that form the alphabet A^2. Then is easy to see that

$$B((0,0),1) = \{(0,0),(0,1),(1,0)\}$$
$$B((1,1),1) = \{(1,1),(0,1),(1,0)\}$$

their intersection is $\{(0,1)(1,0)\}$ thus $\rho(\mathcal{C}) = 0$ and $\tau(\mathcal{C}) = 1$. □

As the balls of radius $\tau(\mathcal{C})$ with center the codewords of $\mathcal{C}$ cover all A^n, the ones of radius $\tau(\mathcal{C}) + 1$ are not pairwise disjoint. Therefore, $\rho(\mathcal{C}) \leq \tau(\mathcal{C})$.

DEFINITION 1.15. A block code $\mathcal{C}$ is said to be *a perfect code* if it is not trivial and $\rho(\mathcal{C}) = \tau(\mathcal{C})$.

When that happens it means that for every possible word $\mathbf{x} \in A^n$, there is a unique codeword $\mathbf{u} \in \mathcal{C}$ in which at most $r = \rho(\mathcal{C}) = \tau(\mathcal{C})$ digits of $\mathbf{u}$ differ from the corresponding digits of $\mathbf{x}$.

As has been said, the detecting and correcting capabilities of a code $\mathcal{C}$ are increased relatively to its minimum distance $\delta(\mathcal{C})$. This is the reason why it is better to have codes whose minimum distance is as large as possible. Another point of interest is to work with codes whose length is small since it will increase the velocity of the transmission. And last but not least, in order to be able to encode the widest variety of messages as possible, it is of interest that the size of the code is large. All this together means that it is of better interest to work with $(n, M, \delta)_q$-codes whose n is as small as possible and its M and δ the greatest possible. But there are some difficulties as

- Small length $n \Rightarrow$ small size since $M \leq |A|^n$.
- Large size $M \Rightarrow$ small minimum distance δ.

The main coding theory problem is to optimize one of the parameters n, M, δ for given values of the other two. The most common problem is to fix q (the number of symbols in the alphabet), n and δ and find the greatest value of M such that an

(n, M, δ)-code exists. This value will be called

$$\mathcal{A}_q(n, \delta) = \{\text{greatest } M \in \mathbb{Z} \mid (n, M, \delta)_q\text{-code exists}\}$$

with $q, n, \delta \in \mathbb{Z}$ such that $q \geq 1$, $0 \leq \delta \leq n$.

DEFINITION 1.16. An (n, M, δ)-code $\mathcal{C}$ is said to be *optimal* if $M = \mathcal{A}_q(n, \delta)$.

THEOREM 1.17. *(Singleton bound) Let q, δ, n be integers such that $n \geq \delta \geq 1$. Then the Singleton bound states,*

$$\mathcal{A}_q(n, \delta) \leq q^{n-\delta+1}.$$

PROOF. Let $\mathcal{C}$ be an optimal (n, M, δ)-code, then $M = \mathcal{A}_q(n, \delta)$. The minimum distance between any two different codewords $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ is δ, therefore they differ from each other in at least one of its coordinates after the $\delta - 1$ first ones. So, if we remove the first $\delta - 1$ coordinates of each codeword of $\mathcal{C}$, a new code $\mathcal{C}' \subseteq A^{n-(\delta-1)}$ with the same size $M' = M = \mathcal{A}_q(n, \delta)$ is obtained. And then,

$$\mathcal{A}_q(n, \delta) = M' \leq |A^{n-\delta+1}| = q^{n-\delta+1}.$$

□

As it has been said previously the aim of a proper communication process is to be able to reproduce the broadcast information from the received one. But unfortunately some bits of information can get lost in the transmission due to the noise in the communication channel.

DEFINITION 1.18. Given a received word $\mathbf{y} \in A^n$ *maximum likelihood decoding* picks a codeword $\mathbf{v} \in \mathcal{C}$ to maximize:

$$\mathbb{P}(\mathbf{y} \text{ received} \mid \mathbf{v} \text{ sent})$$

that is, choose the codeword $\mathbf{v}$ that maximizes the probability that $\mathbf{y}$ was received, given that $\mathbf{v}$ was sent.

DEFINITION 1.19. Given a received word $\mathbf{y} \in A^n$ *minimum distance decoding*, also known as *nearest neighbour decoding*, picks a codeword $\mathbf{v} \in \mathcal{C}$ to minimize the Hamming distance:

$$d(\mathbf{y}, \mathbf{v}) = \#\{i \in [n] \mid y_i \neq v_i\}$$

that is, choose the codeword $\mathbf{v}$ which is as close as possible to $\mathbf{y}$.

THEOREM 1.20. *Let the probability of error p be strictly less than one half and $d = d(\mathbf{y}, \mathbf{v})$, then minimum distance decoding is equivalent to maximum likelihood decoding.*

PROOF.

$$\mathbb{P}(\mathbf{y} \text{ received} \mid \mathbf{v} \text{ sent}) = (1-p)^{n-d} \cdot p^d = (1-p)^n \cdot \left(\frac{p}{1-p}\right)^d$$

which (since p is less than one half) is maximized by minimizing d. □

We will consider a decoding algorithm for linear codes in Chapter 4 and prior to that linear codes will be introduced in Chapter 3. Firstly, however, we need some basic algebraic objects and this is the purpose of Chapter 2.

Chapter 2
Algebraic preliminaries

Although it is assumed that the reader knows the basic notions of both linear and abstract algebra, as well as in working with algebraic structures, in this chapter, with the intention to ease the reader's understanding of the rest of the text, a brief summary of some important algebraic concepts that would be used further on is given.

DEFINITION 2.1. A set R equipped with two internal operations $+$ and $\cdot$, called sum and product, is a *ring* $R = (R, +, \cdot)$, if satisfies the following axioms:

- $(R, +)$ is an abelian group under addition, meaning:
  - $(a + b) + c = a + (b + c)$ $\forall a, b, c \in R$ ($+$ is associative).
  - There is an element $0 \in R$ such that $a + 0 = a$ and $0 + a = a$ $\forall a \in R$ (0 is the additive identity).
  - For each $a \in R$ there exists $-a \in R$ such that $a + (-a) = (-a) + a = 0$ ($-a$ is the additive inverse of a).
  - $a + b = b + a$ for all $a, b \in R$ ($+$ is commutative).
- $(ab)c = a(bc)$ $\forall a, b, c \in R$ ($\cdot$ is associative).
- $a(b + c) = (ab) + (ac)$ $\forall a, b, c \in R$ ($\cdot$ distributive respect to the sum).

A ring R is a *ring with identity* if there is an element denoted by 1 in R such that $a1 = a$ and $1a = a$ (1 is the multiplicative identity). If the product is also commutative, then it is a *commutative ring*.

As in this text we will always work with *commutative rings with identity* whenever we talk about *rings* they should be understood as commutative rings with identity.

DEFINITION 2.2. An element $a \in R$ is a *unit* if it has an inverse, that is if there exists $u^{-1} \in R$ such that $uu^{-1} = 1$. The *set of units* will be denoted by $\mathcal{U}(R)$.

DEFINITION 2.3. A ring is called a *field* $\mathbb{F}$ when every element $a \in R$, except for the additive identity, has an inverse $a^{-1} \in R$ with the multiplication. In other words, if every nonzero element of R is a unit. One can also talk about commutative or noncommutative fields, but in this text we will always work with commutative ones.

REMARK 2.1. Our particular interest in this text is *finite fields*. That is, fields where $\mathbb{F}$ is a finite set, that will be denoted by $\mathbb{F}_q$ where q is the number of elements.

REMARK 2.2. If $\mathbb{F}_q$ is a finite field then $q = p^l$ where p is a prime number and l an integer $l \geq 1$.

EXAMPLE 2.1. Observe that in the set of integers with the familiar addition and multiplication not every element has an inverse under multiplication. Therefore, $\mathbb{Z}$ is a ring whereas $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$ are fields. □

EXAMPLE 2.2. Let $\mathbb{F}$ be a field. The set denoted by $\mathbb{F}[x]$ of all polynomials in the indeterminate x, with form

$$f = a_0 + a_1 x + ... + a_n x^n,$$

where n can be any nonnegative integer and where the coefficients $a_0, a_1, ..., a_n$ are all in $\mathbb{F}$ is a ring called *the polynomial ring* $\mathbb{F}[x]$ whose units are all not null polynomials of degree 0, so $\mathcal{U}(\mathbb{F}[x]) = \mathcal{U}(\mathbb{F}) = \mathbb{F} \setminus \{0\}$. □

DEFINITION 2.4. A ring $I = (I, +, \cdot)$ where $I \subseteq R$ is a subset of R is called an *ideal* of the ring if it satisfies:

- $(I, +)$ is a commutative group.
- For all $a \in R$, and for all $x \in I$, $ax \in I$.

As we assume R to be a commutative ring, $ax = xa$.

DEFINITION 2.5. The *ideal generated by the set* $S = \{s_1, ..., s_n\}$ of R, denoted $\langle S \rangle$ or $(s_1, ..s_n)$, is the smallest ideal of R containing S. That is the intersection of all the ideals $I \in R$ that contain S.

$$\langle S \rangle = \bigcap_{I \supseteq S} = \{a_1 s_1 + \cdots + a_n s_n | a_i \in R, s_i \in S, i = 1, ..., n\}$$

which is, finite linear combinations of the elements of S.

DEFINITION 2.6. We are now going to define R/I called the *quotient ring* R *mod* I for a given a ring R and an ideal $I \subseteq R$. We may define the natural *equivalence relation* $\sim$ *on* R as follows:

$$a \sim b \Leftrightarrow a - b \in I, \quad \forall a, b \in R$$

That in effect is an equivalence relation because it is:

- Reflexive ($a \sim a$, $\forall a \in I$). True since $a \sim a \Leftrightarrow a - a \in I$ and $a - a = 0$ that belong to every ideal.
- Symmetric ($a \sim b \Rightarrow b \sim a$). True since $a \sim b \Rightarrow a - a \in I$ and for each element in I exist an additive inverse $-(a - b) = (b - a)$ is also in I so $b \sim a$.
- Transitive ($a \sim b$ and $b \sim c \Rightarrow a \sim c$, $\forall a, b, c \in I$) True because $a \sim b \Rightarrow a - b \in I$ and $b \sim c \Rightarrow b - c \in I$ so $a - c = a - b + b - c \in I$ and therefore $a \sim c$.

Hence, the *residue class of* a *mod* I for $a \in R$ is:

$$\overline{a} = [a] = \{b \in R | a \sim b\} = \{b \in R | a - b \in I\} = a + I$$

The *quotient ring* R *mod* I is defined as $R/I = \{\overline{a} | a \in R\}$.

PROPOSITION 2.7. *Given a ring* R *and an ideal* $I \subseteq R$*, the equivalence relation mod* I *is well-defined with the sum and product, and therefore, the quotient set* R/I *is a ring.*

PROOF. As the sum in R is an abelian group, as a subset, I is normal, and therefore the equivalence relation is compatible with the sum. For the product, let us consider,

$$a_1 \sim b_1 \quad a_2 \sim b_2$$

that means that there exist $x_1, x_2 \in I$ such that

$$a_1 = b_1 + x_1 \quad a_2 = b_2 + x_2$$

and we want to see if it is compatible with the product, so that $a_1 a_2 \sim b_1 b_2$. Using the what we know,

$$a_1 a_2 = (b_1 + x_1)(b_2 + x_2) = b_1 b_2 + b_1 x_2 + b_2 x_1 + x_1 x_2$$

so $a_1 a_2 - b_1 b_2 \in I$ because $x_1 x_2 \in I$ and also $b_1 x_2, b_2 x_1 \in I$ because I is an ideal. □

DEFINITION 2.8. A *proper ideal* J of R is an ideal of R such that J is a proper subset of R. That is, such that $J \subseteq R$ and $J \neq R$.

DEFINITION 2.9. An ideal $\mathfrak{m} = (m, +, \cdot)$ is said to be a *maximal ideal* if is a maximal element in the set of proper ideals of R. That is, if $\mathfrak{m} \neq R$ and for all I ideal of R such that $m \subseteq I$ it follows that $m = I$ or $I = R$.

PROPOSITION 2.10. *If $\mathfrak{m}$ is a maximal ideal of a ring R then $R/\mathfrak{m}$ is a field.*

PROOF. As $\mathfrak{m}$ is a maximal ideal, $R/\mathfrak{m} \neq 0$. If it was then we would have that $\overline{1} = \overline{0} \Rightarrow 1 - 0 \in \mathfrak{m} \Rightarrow 1 \in \mathfrak{m}$ which implies that $\mathfrak{m} = R$ and that is a contradiction because we have supposed $\mathfrak{m}$ to be maximal so $\mathfrak{m} \neq R$ by definition.

By 2.7 proposition $R/\mathfrak{m}$ is a ring thus in order to be a field we only need to show that every not null element $\overline{a} \in \mathbb{Z}/\mathfrak{m}$ has an inverse. As $\overline{a} \neq 0$ we have that $a \notin \mathfrak{m} \Rightarrow \mathfrak{m} \subsetneq (\mathfrak{m}, a)$ as $\mathfrak{m}$ is maximal $(\mathfrak{m}, a) = R$ so $1 \in R = (\mathfrak{m}, a) \Rightarrow 1 = u + ab$, where $u \in \mathfrak{m}$ and $b \in R$ so $\overline{1} = \overline{u} + \overline{ab} = \overline{0} + \overline{ab} = \overline{a}\ \overline{b}$ and that means $\overline{a}$ has inverse as we wanted to prove. □

DEFINITION 2.11. The degree of a polynomial f, denoted $deg(f)$ is the largest k such that the coefficient of x^k is not zero.

DEFINITION 2.12. A polynomial $p \in \mathbb{F}[x]$ of degree greater or equal than 1 is said to be *irreducible* over $\mathbb{F}$, if it cannot be decomposed, in $\mathbb{F}[x]$, as a product of two polynomials whose degree is less than the degree of f. That is, if $p = fg$, then $deg(f) = deg(p)$ or $deg(g) = deg(p)$.

Note that the subgroups of $\mathbb{Z}$ of the form $n\mathbb{Z}$ are ideals and therefore by Proposition 2.7 the following definition can be given.

DEFINITION 2.13. $\mathbb{Z}/n\mathbb{Z}$ is a ring called the *ring of integers modulo n* or the *ring of integers mod $n\mathbb{Z}$* that has exactly n elements,

$$\mathbb{Z}/n\mathbb{Z} = \{\overline{0}, \overline{1}, ..., \overline{n-1}\}.$$

normally when no confusion can happen the elements in $\mathbb{Z}/n\mathbb{Z}$ will be denoted just by a instead of $\overline{a}$.

PROPOSITION 2.14. (Bezout's identity). *Let a and b be nonzero integers and let d be their greatest common divisor. Then there exist integers x and y such that*

$$ax + by = d.$$

PROPOSITION 2.15. *$\mathbb{Z}/p\mathbb{Z}$ is a field if p is prime.*

PROOF. As $p\mathbb{Z}$ is an ideal by 2.7 it is already know that $\mathbb{Z}/p\mathbb{Z}$ is a ring. Thus in order to be a field we only have to proof that every element $\overline{a} \in \mathbb{Z}/p\mathbb{Z}$, such that $1 \leq a \leq p-1$, has inverse. The fact that p is prime implies that a and p are relatively prime. From Bezout's identity (Proposition 2.14) it follows that there are integers x, y such that $xa + yp = 1$. Let r be the remainder when you divide x by p. That is, write $x = np + r$ where $0 \leq r \leq p$. Then $ra + (na + y)p = 1$. It follows that $\overline{r}\ \overline{a} = \overline{1}$ so $\overline{a}$ has inverse and hence, $\mathbb{Z}/p\mathbb{Z}$ is a field if p is prime. □

PROPOSITION 2.16. *Let $\mathbb{F}$ be a field and f a polynomial in $\mathbb{F}[x] \setminus \{0\}$. If f is irreducible then the ideal (f) is maximal so $\mathbb{F}[x]/(f)$ is a field.*

PROOF. Note the last implication follows from Proposition 2.10. Therefore let us see that effectively if f is irreducible then the ideal (f) is maximal. Suppose (f) is not maximal, then $(f) \subsetneq (a)$ for some $a \in \mathbb{F}[x]$ and such that $(a) \neq \mathbb{F}[x]$. Thus $f = ab$ with $deg(b) > 0$ which implies that f is not irreducible. □

DEFINITION 2.17. We will say that $\mathbb{F}$ is a *finite field* $\mathbb{F}_q$ or *Galois field* $GF(q)$ if it is a field with a finite number q of elements.

THEOREM 2.18. *(Fermat's little theorem) Let $\mathbb{F}_q$ be a finite field of order q. For every nonzero $a \in \mathbb{F}_q$, $a^{q-1} = 1$.*

PROOF. Consider the product of all nonzero elements of $\mathbb{F}_q$, thus:

$$u = \prod_{0 \neq x \in \mathbb{F}_q} x \in \mathbb{F}_q. \tag{2.1}$$

Note that this product has $q-1$ factors. Since the field elements commute, the value of the product u does not depend on the order in which the elements of $\mathbb{F}_q$ are multiplied. In particular, the map

$$\begin{aligned} m\colon \mathbb{F}_q &\to \mathbb{F}_q \\ x &\mapsto m(x) = ax \end{aligned}$$

is bijective for any non-zero $a \in \mathbb{F}_q$ (its inverse is the map $m^{-1}\colon x \mapsto a^{-1}x$) so if in (2.1) x is substituted by $m(x) = ax$, only the factors are permuted but the value of u does not change. So

$$u = \prod_{0 \neq x \in \mathbb{F}_q} x = \prod_{0 \neq x \in \mathbb{F}_q} (ax) = a^{q-1} \prod_{0 \neq x \in \mathbb{F}_q} x = a^{q-1}u \in \mathbb{F}_q.$$

Moreover $u \neq 0$ since it is a product of non-zero field elements so $(a^{q-1} - 1)u = 0 \Rightarrow a^{q-1} - 1 = 0$ to obtain that $a^{q-1} = 1$ for any non-zero element of a finite field. □

LEMMA 2.19. *Let $\mathbb{F}_q$ be a finite field of order $q > 2$ and $k \in \{0, 1, 2, \ldots, q-1\}$. Then*

$$\sum_{a \in \mathbb{F}_q} a^k = \begin{cases} 0 & \text{if } k \in \{0, 1, 2, \ldots, q-2\}; \\ -1 & \text{if } k = q-1. \end{cases}$$

PROOF. Let us first suppose $k \in \{0, 1, 2, \ldots, q-2\}$ and consider the sum of k-th powers of the field elements:

$$S_k = \sum_{x \in \mathbb{F}_q} x^k.$$

For every nonzero value $a \in \mathbb{F}_q$, the map $m\colon x \mapsto ax$ is bijective (just as in the proof of Theorem 2.18) so substituting x by $m(x) = ax$ in the definition of S_k merely permutes the terms in the sum. Thus

$$S_k = \sum_{x \in \mathbb{F}_q} x^k = \sum_{x \in \mathbb{F}_q} (ax)^k = a^k \sum_{x \in \mathbb{F}_q} x^k = a^k S_k$$

for all nonzero values of $a \in \mathbb{F}_q$. If $S_k \neq 0$ cancellations can be done to obtain $a^k = 1$ for all nonzero $k \in \mathbb{F}_q$. This would imply that a polynomial $x^k - 1 \in \mathbb{F}_q[x]$ would have $q-1$ distinct roots in $\mathbb{F}_q$. This cannot happen since the polynomial $x^k - 1$ has degree k that is by assumption less or equal than $q-2$. Therefore it must be that $S_k = 0\ \forall k \in \{0, 1, \ldots, q-2\}$.

Finally let us consider the case $k = q-1$. In this case,

$$S_{q-1} = \sum_{x \in \mathbb{F}_q} x^q - 1 = q - 1$$

since $0^{q-1} = 0$, whereas the remaining $q-1$ terms in the sum all have value 1 by Fermat's little theorem (Theorem 2.18). Note that the right hand side $q-1$ must be interpreted as an element of $\mathbb{F}_q$ so it can be simply identified with -1. □

DEFINITION 2.20. A *Galois ring* $GR(p^m, r)$ is a ring of the form $(\mathbb{Z}/p^m\mathbb{Z})[x]/(f)$, where p is prime, m an integer, $f \in (\mathbb{Z}/p^m\mathbb{Z})[x]$ is a monic polynomial of degree r which is irreducible modulo p and (f) is the ideal of $(\mathbb{Z}/p^m\mathbb{Z})[x]$ generated by f.

Generally, in coding theory, the alphabet set is supposed to be a finite set of integers therefore in this text we will mainly focus on the following next two situations:

DEFINITION 2.21. When $r = 1$, then $GR(p^m, 1) = \mathbb{Z}/p^m\mathbb{Z}$ in this text when we talk about a *Galois ring* or *p-adic ring* we will refer to them and we will denote them $GR(q)$ where $q = p^m$.

DEFINITION 2.22. When $m = 1$, then $GR(p, r) = (\mathbb{Z}/p\mathbb{Z})[x]/(f)$, as p is prime by Proposition 2.15 $\mathbb{Z}/p\mathbb{Z}$ is a finite ring of p elements and as f is irreducible by Proposition 2.16 $GR(p, r) = \mathbb{F}_p[x]/(f)$ is a field, particularly a *finite* or *Galois field* of p^r elements ($r = deg(f)$). By definition $deg(f) = r$ so we will normally refer to them as $\mathbb{F}_q$ for $q = p^r$ or by $GF(p^r)$ when we want to draw attention to the value of p as it is the cardinality of the field to which the coefficients belong.

EXAMPLE 2.3. Even though $GR(4)$ and $\mathbb{F}_4$ both have 4 elements, they are not isomorphic.

- By Definition 2.21, $GR(4) = \mathbb{Z}/4\mathbb{Z} = \{0, 1, 2, 3\}$ thus its addition and multiplication tables are:

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

- By Definition 2.22, $\mathbb{F}_4 = GF(2^2) = \mathbb{F}_2[x]/(f)$, so $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$ and f can be any monic irreducible polynomial in $\mathbb{F}_2$. The polynomial $f = 1 + x + x^2$ is irreducible in $\mathbb{F}_2$ because $f(0) = 1$ and $f(1) = 1$ so it has no roots. Let us assume $\mathbb{F}_4 = \{0, 1, x, 1 + x\}$ thus it's addition and multiplication tables are:

$+$	0	1	x	$1+x$
0	0	1	x	$1+x$
1	1	0	$1+x$	x
x	x	$1+x$	0	1
$1+x$	$1+x$	x	1	0

$\cdot$	0	1	x	$1+x$
0	0	0	0	0
1	0	1	x	$1+x$
x	0	x	$1+x$	1
$1+x$	0	$1+x$	1	x

As can be seen for both $GR(4)$ and $\mathbb{F}_4$ there is symmetry in the table of both operations this is because we are working with commutative structures. The 0 entries in the both sum tables indicate that all elements in both structures have an opposite element for the sum, but as has been pointed out, in $\mathbb{F}_4$ we can see a one entry in the multiplication table for each element of the field while in $GR(4)$ not. This is because $GR(4)$ in a ring and not a field and therefore not every element has an inverse. □

Definition 2.23. Let R be a ring. A *left module M over R* or *left R-module M* consists of a set M with two operations $+ : M \times M \to M$ and $\cdot : R \times M \to M$ denoted addition and scalar multiplication satisfying:

- $(M, +)$ is an abelian group.
- M with the scalar multiplication satisfies:
  (1) $(a + b)\mathbf{x} = a\mathbf{x} + b\mathbf{x},\ \forall a, b \in R,\ \mathbf{x} \in M$;
  (2) $a(\mathbf{x} + \mathbf{y}) = a\mathbf{x} + a\mathbf{y},\ \forall a \in R,\ \mathbf{x}, \mathbf{y} \in M$;
  (3) $a(b\mathbf{x}) = (ab)\mathbf{x},\ \forall a, b \in R,\ \mathbf{x} \in M$;
  (4) $1\mathbf{x} = \mathbf{x},\ \forall \mathbf{x} \in M$.

A *right R-module M* is defined similarly, except that the ring acts on the right; i.e, scalar multiplication takes the form $\cdot : M \times R \to M$, and the above axioms are written with scalars a and b on the right of x and y. But as we will work with rings R which are commutative rings with identity, left R-modules are the same as right R-modules and will be simply called *R-modules*.

Remark 2.3. In the special case that R is a field $\mathbb{F}$ in Definition 2.23 M is called a *vector space over* $\mathbb{F}$ or *$\mathbb{F}$-vector space* and will be denoted by V.

Definition 2.24. Let M an R-module. A *submodule of M* is a subset $N \subset M$ that, with the sum and scalar multiplication of M, is itself an R-module. That is, it satisfies:

(1) If $\mathbf{x}, \mathbf{y} \in N$, then $\mathbf{x} + \mathbf{y} \in N$,

(2) If $\mathbf{x} \in N$ and $a \in R$, then $a\mathbf{x} \in N$.

In other words, W is closed under addition of vectors and under scalar multiplication.

REMARK 2.4. In the special case that R is a field $\mathbb{F}$ in Definition 2.24 M is a vector space denoted V (Remark 2.3) and therefore N is called a *vector subspace of* V and will be denoted by W.

DEFINITION 2.25. A *basis for an R-module M* is a linearly independent generating set. In other words, for an R-module M, a set $E \subseteq M$ is a basis for M if,

- E is a generating set for M which means for modules that, every element of M is a finite sum of elements of E multiplied by coefficients in R;
- E is *linearly independent over* R, that is, if $\sum_{i=1}^{n} a_1 e_i = 0_M$ for $e_1, e_2, ..., e_n$ distinct elements of E, then $a_i = 0_R$ for all $i \in \{1, 2, ..., n\}$ (here we have used 0_M to denote the zero element of M and 0_R the zero element of R).

DEFINITION 2.26. An R-module M is said to be a *free module* if it has a basis.

REMARK 2.5. The term free module extends to all modules which are isomorphic to R^n.

REMARK 2.6. Not all modules are free.

EXAMPLE 2.4. The module $\mathbb{Z}/m\mathbb{Z}$ for any integer $m > 1$ is mod

- free as a module over itself;
- not free as a $\mathbb{Z}$-module. Having m elements cannot be isomorphic to any of the modules $\mathbb{Z}^n$, which are all infinite sets.

DEFINITION 2.27. A *free submodule* is a submodule of a module which is free as a module.

REMARK 2.7. By remark 2.5 a free submodule is a submodule isomorphic to R^k for some k.

Any two basis have the same cardinality, then,

DEFINITION 2.28. The *rank of a free module* is the cardinality of any (and therefore every) basis.

Chapter 3
Linear codes over fields

In order to define useful structures on codes, a first step is to enrich the alphabet A with more structure than being merely a set. If the aim, for example, is to be able to sum two words, a group could be used instead of A so the letters will be elements of the group and therefore the addition of two words will be defined as the sum of their letter coordinate by coordinate with usual properties. Moreover if instead of a group we use a ring, it will also be possible to multiply words by alphabet symbols in order to obtain another word in in the ring. If the aim is to be able to divide symbols, then a field $\mathbb{F}$ can be used so all elements are units. In this chapter we will introduce linear codes over fields which are subspaces of a vector space over a field while in Chapter 7 we will consider linear codes will over rings which are submodules of a module over a ring R. Therefore in Chapter 7 not all symbols will have an inverse so things will not work as smoothly and more attention will have to be paid when working with them. For example, when working with the integers quotient ring $\mathbb{Z}/q\mathbb{Z}$, division by nonzero elements is possible without exception if and only if q is prime.

DEFINITION 3.1. A *$[n,k]_q$-linear-code* $\mathcal{C}$ which is called a *linear code* of length n and rank k is a linear subspace $\mathcal{C}$ of dimension k of the vector space $\mathbb{F}_q^n$ where $\mathbb{F}_q$ is a finite field of order q, for some prime power q ($q = p^e$). The vectors in $\mathcal{C}$ are the codewords.

One can specify a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ by giving a basis $\mathbf{u}_1, ..., \mathbf{u}_k$ for $\mathcal{C}$, so the codewords $\mathbf{u} \in \mathcal{C}$ are the linear combinations

$$\lambda_1 \mathbf{u}_1 + \cdots + \lambda_k \mathbf{u}_k \text{ with } \lambda_i \in \mathbb{F}_q$$

of the basis vectors. The fact that $\mathbf{u}_1, ..., \mathbf{u}_k$ is a basis means that there are as many linear combinations as vectors $(\lambda_1, ..., \lambda_k) \in \mathbb{F}_q^k$, which is, q^k. Hence, a $[n,k]_q$-linear-code has size equal to q^k.

One of the main advantages of using linear codes is that in order to specify the code only k vectors need to be given (where $k = \dim(\mathcal{C})$), rather than all $M = q^k$ vectors in $\mathcal{C}$. There are more consequences of a code being linear, one is that, since it is a linear subspace, the sum or difference of two codewords is always another codeword and the zero vector is always a codeword.

In a linear code the minimum distance $\delta(\mathcal{C})$ can be determined by just looking at the distance of each codeword to the word $\mathbf{0}$.

PROPOSITION 3.2. *In a linear code the minimum distance is equal to the minimum weight among all non-zero codewords. In other words, if $\mathcal{C}$ is a $[n,k]_q$-linear-code, then*

$$\delta(\mathcal{C}) = \min\{w(\mathbf{u}) \mid \mathbf{0} \neq \mathbf{u} \in \mathcal{C}\}$$

PROOF. On one side, $w(\mathbf{u}) = d(\mathbf{u}, \mathbf{0}) \geq \delta(\mathcal{C})$ for every $\mathbf{u} \in \mathcal{C}$ not null. So $\delta(\mathcal{C}) \leq \min\{w(\mathbf{u}) \mid \mathbf{0} \neq \mathbf{u} \in \mathcal{C}\}$. On the other side, given $\mathbf{v}, \mathbf{w} \in \mathcal{C}$ so that $\delta(\mathcal{C}) = d(\mathbf{v}, \mathbf{w})$. Since $\mathcal{C}$ is linear,

$$\delta(\mathcal{C}) = d(\mathbf{v}, \mathbf{w}) = w(\mathbf{v} - \mathbf{w}) \geq \min\{w(\mathbf{u}) \mid \mathbf{0} \neq \mathbf{u} \in \mathcal{C}\}$$

Hence, the equality is true. □

If the minimum distance of a $[n,k]_q$-linear-code is δ, it will be said that it is a $[n,k,\delta]$-linear-code and as we have already said, for a linear code $M = q^k$ so it is a (n, q^k, δ)-code.

In general, finding the minimum distance of a code requires comparing every pair of distinct elements of the code which means that if $\mathcal{C}$ is a code without structure of size M, in order to determine $\delta(\mathcal{C})$ one is required to calculate

$$\binom{M}{2} = \frac{M(M-1)}{2}$$

distances. While in a linear code a lower number of comparisons are needed. As seen in the previous proposition it is enough to calculate just the $M-1$ weights of the non-zero codewords.

It is well known that two natural ways of describing a vector subspace are by giving its basis, or the basis of its orthogonal subspace with respect to a given inner product. Therefore, there are also two natural ways for describing a linear code.

DEFINITION 3.3. Let $\mathcal{C}$ be a $[n,k]_q$-linear-code. A *generator matrix* G of $\mathcal{C}$ is a matrix G which has as its rows a set of basis vectors of the linear subspace $\mathcal{C}$. Therefore, G is a $k \times n$ matrix, $G \in \mathbb{F}_q^{k \times n}$.

Conversely, given a matrix $G \in \mathbb{F}_q^{k \times n}$ the subspace $\langle G \rangle \subseteq \mathbb{F}_q^n$ generated by the rows of G is a code of type $[n,k]$, where k is the rank of G. It is then said that $\langle G \rangle$ is the *code generated by G*.

The code $\mathcal{C}$ is the set of all linear combinations of the rows of G, or as it is usually called, the row space of G.

Given the matrix G, the code $\mathcal{C}$ is obtained by multiplying G on the left by all possible $1 \times k$ row vectors (this gives all possible linear combinations, thus, the code):

$$\mathcal{C} = \{\mathbf{u}G | \mathbf{u} \in \mathbb{F}_q^k\}.$$

EXAMPLE 3.1. Let $G = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \in \mathcal{M}_{2\times 3}(\mathbb{Z}_2)$. As the rows are linearly independent, it has rank 2 and generates a binary linear code $\mathcal{C}$ with parameters $[3, 2]$. Since

$$(x_1, x_2)\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = (x_1, x_1 + x_2, x_2)$$

the words 00, 01, 10 and 11 are coded as follows

$$00 \to 000, \quad 01 \to 011, \quad 10 \to 110, \quad 11 \longrightarrow 101$$

Then $\mathcal{C} = \{000, 011, 101, 110\}$ and has minimum distance 2. □

Given a generator matrix G for a linear code $\mathcal{C}$, it can be quite tedious to determine whether a received word $\mathbf{x}$ belongs to the code or not, and if not, which element $\mathbf{u} \in \mathcal{C}$ is closest to it. Because in order to find this out, using the generator matrix, one would be required to calculate all the codewords and compare them with the received vector. To make this easier, an alternative matrix description of $\mathcal{C}$ can be given. It consist on giving a set of $n-k$ simultaneous linear equations which define the elements of $\mathcal{C}$, so that a vector belongs to the code if and only if it satisfies these equations.

DEFINITION 3.4. The *dual code* $\mathcal{C}^\perp$ of a $[n, k]_q$-linear-code $\mathcal{C}$ is the $[n, n-k]_q$-linear-code which is the orthogonal subspace $\mathcal{C}^\perp$ of $\mathcal{C}$. It is well defined since an $[n, k]_q$-linear-code $\mathcal{C}$ is a vector subspace of $\mathbb{F}_q^n$ of dimension k and hence its orthogonal subspace $\mathcal{C}^\perp$ is a vector subspace of $\mathbb{F}_q^n$ of dimension $n - k$, which is, the dual $[n, n - k]_q$-linear-code of the code.

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{xu} = 0, \forall \mathbf{u} \in \mathcal{C}\},$$

where $\mathbf{xu}$ is the standard inner product,

$$\mathbf{xu} =< \mathbf{x}, \mathbf{u} >= \sum_{i=1}^{n} x_i u_i.$$

A linear code $\mathcal{C}$ is uniquely determined by giving one of its bases (generator matrix) or by giving a basis for the dual code.

DEFINITION 3.5. A generator matrix H of $\mathcal{C}^\perp$ is called a *check matrix* of $\mathcal{C}$.

Then H is the matrix of an homogeneous system of linear equations whose solutions are exactly the vectors of $\mathcal{C}$. Those are the equations that have been mentioned before to define the elements of a code. These equations are very useful when dealing with error detection and correction, which is why the check matrix is more often used than the generator matrix.

LEMMA 3.6. *Let $\mathcal{C}$ be a linear code, a codeword $\mathbf{u}$ belongs to $\mathcal{C}$ if and only if the vector-matrix product $\mathbf{u}H^T$ is equal to 0.*

LEMMA 3.7. *Let $\mathcal{C}$ be a $[n, k]$-code over $\mathbb{F}_q$ with a generator matrix G, and let H be a matrix over $\mathbb{F}_q$ with n columns and $n - k$ rows. Then H is a check matrix for $\mathcal{C}$ if and only if H has rank $n - k$ and satisfies $GH^T = \mathbf{0}$.*

PROOF. On the one hand, the rows of H are $n-k$ vector in $\mathbb{F}_q^n$, and $GH^T = 0$ if and only if these rows are orthogonal to those of G, or equivalently lie in $\mathcal{C}^\perp$. On the other hand, H has rank $n-k$ if and only if its rows are linearly independent, or equivalently form a basis for $\mathcal{C}^\perp$; thus H satisfies the given conditions if and only if it is a generator matrix for $\mathcal{C}^\perp$ and that is exactly, a check matrix for $\mathcal{C}$. □

In general, a vector space does not have a unique basis. For this reason a generator matrix G and a check matrix H of a linear code $\mathcal{C}$ are not generally unique.

EXAMPLE 3.2. Given $(1,1,0,0)$, $(0,1,1,0)$ and $(0,0,1,1)$ three linear independent vectors of $\mathbb{F}_2^4$, the subspace

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

generated by them is a linear code of dimension 3, that is, a $[4,3]_2$-linear-code. The size of the code is $2^3 = 8$, those 8 vectors are all the posible linear combinations, so the code is:

$$\begin{aligned}\mathcal{C} = \{&(0,0,0,0), (1,1,0,0), (0,1,1,0), (0,0,1,1),\\ &(1,0,1,0), (1,1,1,1), (0,1,0,1), (1,0,0,1)\}.\end{aligned}$$

These three vector are a basis for the code $\mathcal{C}$ and a generator matrix can be given as

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

However the vectors $(1,0,0,1)$, $(0,1,0,1)$ and $(0,0,1,1)$ are also linearly independent and form another basis of $\mathcal{C}$. Thus, $\mathcal{C}$ admits also

$$G_2 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

as a generator matrix. The codifications using one or the other are the following ones:

$\mathbf{x} \in \mathbb{F}_2^3$	$\mathbf{x}G_1$	$\mathbf{x}G_2$
$(0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
$(0,0,1)$	$(0,0,1,1)$	$(0,0,1,1)$
$(0,1,0)$	$(0,1,1,0)$	$(0,1,0,1)$
$(1,0,0)$	$(1,1,0,0)$	$(1,0,0,1)$
$(0,1,1)$	$(0,1,0,1)$	$(0,1,1,0)$
$(1,1,0)$	$(1,0,1,0)$	$(1,1,0,0)$
$(1,0,1)$	$(1,1,1,1)$	$(1,0,1,0)$
$(1,1,1)$	$(1,0,0,1)$	$(1,1,1,1)$

Notice that as the first three columns of G_2 form the identity matrix, the first three coordinates of $\mathbf{x}G_2$ are always the vector $\mathbf{x}$ itself, which does not happen when using G_1. In this case, given a word $\mathbf{u} \in \mathcal{C}$ the decoding process, which is finding the solution to the system $\mathbf{x}G = \mathbf{u}$, is immediate for G_2. □

As shown in the example below, sometimes it is more useful to choose one basis than another since calculations become easier. The rows $\mathbf{r}_1, ..., \mathbf{r}_k$ of G, regarded as elements of $\mathbb{F}^n$, form a basis of $\mathcal{C}$ so elementary row operations:

- **Row switching:** ($\mathbf{r}_i \leftrightarrow \mathbf{r}_j$)
  A row within the matrix can be switched with another row.
- **Row multiplication:** ($k\mathbf{r}_i \to \mathbf{r}_i, k \neq 0$)
  A row can be multiplied by a non-zero constant .
- **Row addition:** ($\mathbf{r}_i + k\mathbf{r}_j \to \mathbf{r}_i, i \neq j$)
  A row can be replaced by the sum of that row and a multiple of another row.

can be used. When applying them the basis for $\mathcal{C}$ may change but not the subspace $\mathcal{C}$ spanned by the rows. Thus, elementary row operations can be applied to any given generator matrix in order to find one with better properties without changing the code generated by it.

PROPOSITION 3.8. *Let G_1 and G_2 be two matrices of $\mathbb{F}_q^{k\times n}$ with the k rows linearly independent. Hence, G_1 and G_2 are generator matrices of the same code if and only if G_2 can be obtained from G_1 by elementary row operations.*

PROOF. Suppose that G_2 is obtained from G_1 by elementary row operations. The rows of G_1 form a basis of a code $\mathcal{C}$. Applying elementary row operations to a basis of a subspace changes it into another basis of the same subspace. Hence the rows of G_2 also form a basis of the same code $\mathcal{C}$ and thus, G_2 is a generator matrix of $\mathcal{C}$.

Reciprocally, suppose that G_1 and G_2 are generator matrices of the same code $\mathcal{C}$. Calling $\mathbf{u}_1, ..., \mathbf{u}_k$ the rows of G_1 and $\mathbf{v}_1, ..., \mathbf{v}_k$ the rows of G_2. The rows of G_1 form a basis of $\mathcal{C}$ so $\mathbf{v}_1$ is a linear combination of $\mathbf{u}_1, ..., \mathbf{u}_k$:

$$\mathbf{v}_1 = \lambda_1\mathbf{u}_1 + \cdots + \lambda_k\mathbf{u}_k.$$

Since $\mathbf{v}_1 \neq 0$, there exists for some i a coefficient λ_i different from zero. Thus, replacing $\mathbf{u}_i$ for $\mathbf{v}_1$ a new basis $\mathbf{u}_1, ..., \mathbf{u}_{i-1}, \mathbf{v}_1, \mathbf{u}_{i+1}, ..., \mathbf{u}_k$ for $\mathcal{C}$ is obtained. Now by permuting the rows in order to have a new order a new basis $\mathbf{v}_1, \mathbf{w}_2, ..., \mathbf{w}_k$ is obtained. The vector $\mathbf{v}_2$ admits an expression

$$\mathbf{v}_2 = \mu_1\mathbf{v}_1 + \mu_2\mathbf{w}_2 + \cdots + \mu_k\mathbf{w}_k.$$

If $\mu_2 = \cdots = \mu_k = 0$, the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ would be linearly dependent, which can not happen since they both are part of a basis. Hence, for some $i \geq 2$ exists $\mu_i \neq 0$. As before, a new basis $\mathbf{v}_1, \mathbf{v}_2, \mathbf{z}_3..., \mathbf{z}_k$ can be obtained. Repeating this we obtain the basis $\mathbf{v}_1, ..., \mathbf{v}_k$. Therefore, it is posible to pass from G_1 to G_2 with just elementary row operations. □

However, if instead of rows, *columns* of $\mathcal{C}$ are permuted, $\mathcal{C}$ may change. But the new code will differ from it only in the order of symbols within code-words; the two codes will have the same parameters such as n, k, d, M, etc., so they are not essentially different. This motivates the following definition.

DEFINITION 3.9. Two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are *equivalent* if they have generator matrices G_1 and G_2 which differ only by elementary row operations, column permutations and/or multiplying a column by a non-zero scalar.

Informally, one tends to think of $\mathcal{C}_1$ and $\mathcal{C}_2$ as "the same code", even though they generally consist of different code-words.

By systematically using elementary row operations and column permutations, any generator matrix can be converted into the form

$$G = (I_k \mid P) = \begin{pmatrix} 1 & & & & * & * & \cdots & * \\ & 1 & & & * & * & \cdots & * \\ & & \ddots & & \vdots & \vdots & & \vdots \\ & & & 1 & * & * & \cdots & * \end{pmatrix}, \tag{3.1}$$

where I_k is the $k \times k$ identity matrix, and P is a matrix with k rows and $n-k$ columns, represented by the asterisks.

Definition 3.10. It is said that the generator matrix G of a code $\mathcal{C}$ is in *systematic form* when it has form (3.1).

In this case, each word $\mathbf{a} = a_1 \dots a_k \in \mathbb{F}_q^k$ is encoded as

$$\mathbf{u} = \mathbf{a}G = a_1 \dots a_k a_{k+1} \dots a_n,$$

where $a_1, \dots, a_k$ are information digits and $a_{k+1} \dots a_n = \mathbf{a}P$ is a block of $n-k$ check digits.

Proposition 3.11. *Let $\mathcal{C}$ be a $[n,k]_q$-linear-code and $G \in \mathbb{F}_q^{k\times n}$ and $H \in \mathbb{F}_q^{(n-k)\times n}$ two matrices of form*

$$G = (I_k \mid P) \quad \text{and} \quad H = (-P^T \mid I_{n-k}).$$

Hence, G is the systematic generator matrix of $\mathcal{C}$ if and only if H is a check matrix.

Proof. The k rows of G are linearly independents, as well as the $n-k$ of H. Thus,

$$GH^T = (I_k \mid P)(-P^T \mid I_{n-k})^T = -P + P = 0.$$

Then, the $n-k$ rows of H are perpendicular to the k rows of G. So, the rows of G form a basis of $\mathcal{C}$ if and only if the rows of H form a basis of $\mathcal{C}^\perp$. □

Definition 3.12. The uniqueness of the systematic generator matrix implies then the uniqueness of $H = (-P^T \mid I_{n-k})$, that is called *systematic check matrix*. The codes whose generator and check matrices admit a systematic form are called *systematics codes*.

Example 3.3. Consider the $[7,4]_2$-linear-code of size $2^4 = 16$ generated by

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

by applying the following elementary row transformations

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow[r_3 \to r_3 + r_1]{r_2 \to r_2 + r_1} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\xrightarrow[r_4 \to r_4 + r_2]{r_1 \to r_1 + r_2} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$\xrightarrow{r_2 \to r_2 + r_3} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$\xrightarrow{r_3 \to r_3 + r_4} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

its systematic generator matrix in form $G = (I_4 \mid P)$ is obtained

$$G = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \right)$$

and therefore its systematic check matrix satisfying $H = (-P^T \mid I_3)$ is:

$$H = \left(\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right)$$

□

Finally, let us finish this chapter naming two special types of $[n, k]_q$-linear codes. Reed Solomon codes studied in Chapter 6 are a more complex example of one of them.

DEFINITION 3.13. A code $\mathcal{C}$ is called *self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^{\perp}$.

EXAMPLE 3.4. Let us consider $\mathcal{C}$ to be the $[3, 1]$-linear code over $\mathbb{F}_4$ generated by

$$G = (1, x + 1, x).$$

Therefore, as

$$(a)(1, x + 1, x) = (a, ax + a, ax)$$

the words $0,\ 1,\ x$ and $1 + x$ of $\mathbb{F}_4$ are coded as follows

$\mathbf{x} \in \mathbb{F}_4$	$\mathbf{x}H$
0	$(0, 0, 0)$
1	$(1, 1 + x, x)$

$\mathbf{x} \in \mathbb{F}_4$	$\mathbf{x}H$
x	$(x, 1, 1 + x)$
$1 + x$	$(1 + x, x, 1)$

Then our code is exactly $\mathcal{C} = \{(0,0,0),(1,1+x,x),(x,1,1+x),(1+x,x,1)\}$. As G is in systematic form, by Proposition 3.11, it is known that its systematic check matrix can be given by

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

Therefore as H generates $\mathcal{C}^\perp$ by

$$(a,b)\begin{pmatrix} x+1 & 1 & 0 \\ x & 0 & 1 \end{pmatrix} = ((a+b)x+a, a, b)$$

the words $\mathbf{x} \in \mathbb{F}_4^2$ are coded as

$\mathbf{x} \in \mathbb{F}_4^2$	$\mathbf{x}H$	$\mathbf{x} \in \mathbb{F}_4^2$	$\mathbf{x}H$
$(0,0)$	$(\mathbf{0},\mathbf{0},\mathbf{0})$	$(x,0)$	$(1,x,0)$
$(0,1)$	$(x,0,1)$	$(x,1)$	$(\mathbf{1}+\mathbf{x},\mathbf{x},\mathbf{1})$
$(0,x)$	$(x+1,0,x)$	(x,x)	(x,x,x)
$(0,1+x)$	$(1,0,1+x)$	$(x,1+x)$	$(0,x,1+x)$
$(1,0)$	$(x+1,1,0)$	$(1+x,0)$	$(x,1+x,0)$
$(1,1)$	$(1,1,1)$	$(1+x,1)$	$(x,1+x,1)$
$(1,x)$	$(0,1,x)$	$(1+x,x)$	$(\mathbf{1},\mathbf{1}+\mathbf{x},\mathbf{x})$
$(1,1+x)$	$(\mathbf{x},\mathbf{1},\mathbf{1}+\mathbf{x})$	$(1+x,1+x)$	$(1+x,1+x,1+x)$

the code $\mathcal{C}^\perp = \mathbf{x}H$ is composed by the 4^2 codewords in $\mathbb{F}_4^3$ listed in the previous array. Note that all the codeword from $\mathcal{C}$ appear in $\mathcal{C}^\perp$, they have been highlighted. Thus, $\mathcal{C} \subseteq \mathcal{C}^\perp$ so $\mathcal{C}$ is a self-orthogonal code. □

Definition 3.14. A code $\mathcal{C}$ is called *self-dual* if $\mathcal{C} = \mathcal{C}^\perp$; equivalently, if every generator matrix G is also a check matrix.

Example 3.5. Let us consider $\mathcal{C}$ to be the $[4,2]$-linear code over $\mathbb{F}_3 = \mathbb{Z}/3\mathbb{Z}$ generated by

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

As G is in systematic form, by Proposition 3.11, its systematic check matrix H can be given by

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

Therefore, as using G any word $(a,b) \in \mathbb{F}_3^2$ is encoded as

$$(a,b)\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix} = (a,b,a+b,a+2b)$$

and using H

$$(a,b)\begin{pmatrix} 2 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix} = (2(a+b),2a+b,a,2b).$$

Then the 9 words of $\mathbb{F}_3^2$ are coded as follows

$\mathbf{x} \in \mathbb{F}_3^2$	$\mathbf{x}G$	$\mathbf{x}H$
00	$(0,0,0,0)$	$(0,0,0,0)$
01	$(0,1,1,2)$	$(2,1,0,1)$
02	$(0,2,2,1)$	$(1,2,0,2)$
10	$(1,0,1,1)$	$(2,2,1,0)$
11	$(1,1,2,0)$	$(1,0,1,1)$
12	$(1,2,0,2)$	$(0,1,1,2)$
20	$(2,0,2,2)$	$(1,1,2,0)$
21	$(2,1,0,1)$	$(0,2,2,1)$
22	$(2,2,1,0)$	$(2,0,2,2)$

Thus it is clear that all codewords in $\mathcal{C} = \{\mathbf{x}G\}$ are also in its dual code $\mathcal{C}^\perp = \{\mathbf{x}H\}$ and viceversa. Therefore, $\mathcal{C}$ is a self-dual code. □

PROPOSITION 3.15. *There is no self-orthogonal code whose length is bigger than twice its dimension.*

PROOF. By definition an $[n,k]_q$-linear code $\mathcal{C}$ is self-orthogonal if $\mathcal{C} \subseteq \mathcal{C}^\perp$. Then, $|\mathcal{C}| \leq |\mathcal{C}^\perp|$ so $q^n \leq q^{n-k} \Rightarrow k \leq n-k$ so $n \leq 2k$. □

COROLLARY. *Every self-dual code length is twice its dimension.*

Chapter 4
Syndrome decoding

An obvious algorithm for decoding is to make a table consisting of the nearest codeword for each of the q^n possible received words in $\mathbb{F}_q^n$ so that once a word $\mathbf{y}$ is received it can be decoded by looking it up in the table. Despite the ease of this method is obviously impractical if q^n is very large. Notwithstanding, as well as it happened when calculating the minimum distance, working with linear codes has its advantages. So when decoding linear codes there is no need to calculate all distances $d(\mathbf{y}, \mathbf{u})$ with $\mathbf{u} \in \mathcal{C}$ in order to decode a received word $\mathbf{y}$. As it will be explained in this subsection when working with linear codes a similar but more efficient decoding process can be used, it is called *syndrome decoding.*

DEFINITION 4.1. Let $\mathcal{C}$ be an $[n, k]_q$-linear code. For any word $\mathbf{x} \in \mathbb{F}_q^n$, the set

$$\mathbf{x} + \mathcal{C} = \{\mathbf{x} + \mathbf{u} \mid \mathbf{u} \in \mathcal{C}\}$$

is called a *coset* of $\mathcal{C}$.

Every word $\mathbf{y}$ is in some coset (in $\mathbf{y} + \mathcal{C}$ for example) and two diferent words $\mathbf{x}$ and $\mathbf{y}$ are in the same coset if and only if $\mathbf{x} - \mathbf{y} \in \mathcal{C}$. Each coset contains q^k words.

PROPOSITION 4.2. *Two cosets are either disjoint or coincide.*

PROOF. If $(\mathbf{x} + \mathcal{C}) \cap (\mathbf{y} + \mathcal{C}) \neq 0$, there exists $\mathbf{z} \in (\mathbf{x} + \mathcal{C}) \cap (\mathbf{y} + \mathcal{C})$. Then $\mathbf{z} = \mathbf{x} + \mathbf{u} = \mathbf{y} + \mathbf{v}$, where $\mathbf{u}$ and $\mathbf{v}$ belong to $\mathcal{C}$. Therefore $\mathbf{y} = \mathbf{x} + \mathbf{u} - \mathbf{v} = \mathbf{x} + \mathbf{w}$, where $\mathbf{u} - \mathbf{v} = \mathbf{w} \in \mathcal{C}$, and so $\mathbf{x} + \mathcal{C} \subseteq \mathbf{y} + \mathcal{C}$. Similarly $\mathbf{y} + \mathcal{C} \subseteq \mathbf{x} + \mathcal{C}$, and so $\mathbf{x} + \mathcal{C} = \mathbf{y} + \mathcal{C}$ □

Therefore $\mathbb{F}_q^n$ can be partitioned into cosets of $\mathcal{C}$:

$$\mathbb{F}_q^n = \mathcal{C} \cup (\mathbf{x}_1 + \mathcal{C}) \cup (\mathbf{x}_2 + \mathcal{C}) \cup \cdots \cup (\mathbf{x}_t + \mathcal{C}) \tag{4.1}$$

where $t = q^{n-k} - 1$ since the number of cosets is $|\mathbb{F}_q^n / \mathcal{C}| = \frac{|\mathbb{F}_q^n|}{|\mathcal{C}|} = \frac{q^n}{q^k} = q^{n-k}$.

Suppose that a word $\mathbf{y} \in \mathbb{F}_q^n$ is received instead of $\mathbf{v} \in \mathcal{C}$. The received word must belong to some coset in (4.1), say $\mathbf{y} = \mathbf{x}_i + \mathbf{u}$ for some codeword $\mathbf{u}$ and some i. As has been already defined, the error vector is $\mathbf{e} = \mathbf{y} - \mathbf{v}$ therefore $\mathbf{e} = \mathbf{x}_i + \mathbf{u} - \mathbf{v} = \mathbf{x}_i + \mathbf{w}$ where $\mathbf{w} \in \mathcal{C}$ so $\mathbf{e} \in \mathbf{x}_i + \mathcal{C}$ (the same coset as $\mathbf{y}$). Thus it has just been proved that the possible error vectors are exactly the vectors in the coset containing $\mathbf{y}$. This

is the reason why the decoder's strategy is, given $\mathbf{y}$, to choose a minimum weight vector in the coset containing $\mathbf{y}$ in order to use it as the error vector so that $\mathbf{y}$ would be decoded as $\mathbf{v} = \mathbf{y} - \mathbf{e}$.

DEFINITION 4.3. The minimum weight vector in a coset is called the *coset leader*. If there is more than one vector with the minimum weight, one can be chosen at random and called the coset leader.

Note that the $\mathbf{x}_i$ in (4.1) will be supposed to be precisely the coset leaders and that the fact of choosing the error vector as the minimum weight vector responds to the fact that it is generally assumed that during a transmission process it is more likely that a lower number of error occur than a bigger one.

This decoding process can be easily illustrated by using the following table.

DEFINITION 4.4. A *standard array* for an $[n, k]_q$-linear code is a q^{n-k} by q^k array where:

- The first row lists all codewords (with the $\mathbf{0}$ codeword on the extreme left).
- Each row is a coset with the coset leader in the first column.
- The entry in the i-th row and j-th column is the sum of the i-th coset leader and the j-th codeword.

Because each possible word can appear only once in a standard array some care must be taken during construction. A standard array can be created as follows:

(1) List the codewords of C, starting with the codeword $\mathbf{0}$, as the first row.
(2) Choose any word $\mathbb{F}_q^n$ of minimum weight not already in the array. Write this as the first entry of the next row, it will be the coset leader.
(3) Fill out the row by adding the coset leader to the codeword at the top of each column. The sum of the i-th coset leader and the j-th codeword becomes the entry in row i, column j.
(4) Repeat steps 2) and 3) until all rows (all cosets) are listed and each vector of $\mathbb{F}_q^n$ appears exactly once.

EXAMPLE 4.1. The $[4, 2]_3$-linear code with generator matrix

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

is $\mathcal{C} = \{0000, 1012, 0111, 2021, 0222, 1120, 1201, 2210, 2102\}$. Its standard array constructed following the previous procedure is:

message $\mathbf{x}$	00	10	01	20	02	11	12	22	21
coset $\mathbf{x}_0 + \mathcal{C}$	0000	1012	0111	2021	0222	1120	1201	2210	**2102**
coset $\mathbf{x}_1 + \mathcal{C}$	1000	2012	1111	0021	1222	2120	2201	0210	0102
coset $\mathbf{x}_2 + \mathcal{C}$	0100	1112	0211	2121	0022	1220	1001	2010	2202
coset $\mathbf{x}_3 + \mathcal{C}$	0010	1022	0121	2001	0202	1100	**1211**	2220	2112
coset $\mathbf{x}_4 + \mathcal{C}$	0001	1010	0112	2022	0220	1121	1202	2211	2100
coset $\mathbf{x}_5 + \mathcal{C}$	2000	0012	2111	1021	2222	0120	0201	1210	1102
coset $\mathbf{x}_6 + \mathcal{C}$	0200	1212	0011	2221	0122	1020	1101	2110	2002
coset $\mathbf{x}_7 + \mathcal{C}$	0020	1002	0101	2011	0212	1110	1221	2200	2122
coset $\mathbf{x}_8 + \mathcal{C}$	0002	1011	0110	2020	0221	1122	1200	2212	2101

where coset $\mathbf{x}_0 + \mathcal{C}$ is the code itself as $\mathbf{x}_0 = \mathbf{0}$. Hence, the first row is generated by $\mathbf{x}G = \mathcal{C}$.

Note that all $3^4 = 81$ words in $\mathbb{F}_3^4$ appear, divided into the $3^{4-2} = 9$ cosets forming the rows, and the coset leaders are on the first column. □

Therefore the standard array is used for decoding linear codes in the following way. Once a word $\mathbf{y} \in \mathbb{F}_q^n$ is received its position in the array has to be found. Then the decoder decides that the error vector $\mathbf{e}$ is the coset leader $\mathbf{x}_i$ such that $\mathbf{y} \in \mathbf{x}_i + \mathcal{C}$ (the extreme left component of the standard array row to which $\mathbf{y}_i$ belongs). Thus, $\mathbf{y}$ is decoded as the codeword $\mathbf{v} = \mathbf{y} - \mathbf{e}$ (the codeword found in the top of the standard array column to which $\mathbf{y}_i$ belongs).

Example 4.2. Using the same code and the standard array illustrated in 4.1 Example.

- If $\mathbf{y} = 1211$ is received finding it in the table $\mathbf{e} = 0010$ and it is decoded as $\mathbf{v} = 1211 - 0010 = 1201$ (which corresponds to message 12).
- If $\mathbf{y} = 2102$ is received finding it in the table $\mathbf{e} = 0000$ and it is decoded as $\mathbf{v} = 2102 - 0000 = 2102$ (which corresponds to message 21). □

As has been said before despite the simplicity of this method it is not efficient since all q^n words need to be listed in the $q^{n-k} \times q^k$ standard array. The syndrome decoding process follows the same outline but with a reduced $q^{n-k} \times 2$ array.

Definition 4.5. Let $\mathcal{C}$ be a $[n,k]_q$-linear-code with check matrix H. The *syndrome* of a word $\mathbf{x} \in \mathbb{F}_q^n$ (with respect to H) is

$$s(\mathbf{x}) = \mathbf{x}H^T \in \mathbb{F}_q^{n-k}.$$

Proposition 4.6. *Let $\mathcal{C}$ be a $[n,k]_q$-linear-code with check matrix H and $\mathbf{x} \in \mathbb{F}_q^n$ then $s(\mathbf{x})$ satisfies the following properties:*

(1) $s(\mathbf{x})$ is a word of lenght $n-k$.
(2) $s(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ is a codeword.
(3) given another word $\mathbf{y} \in \mathbb{F}_q^n$, $s(\mathbf{x}) = s(\mathbf{y})$ if and only if they belong to the same coset of $\mathcal{C}$.

Proof. (1) By definition H is an $(n-k) \times n$ thus its transpose is an $n \times (n-k)$ matrix and the vector matrix product result is a vector in $\mathbb{F}_q^{n-k}$.
(2) 3.6 Lemma.
(3) For any two diferents words $\mathbf{x}$ and $\mathbf{y}$, $s(\mathbf{x}) = s(\mathbf{y}) \Leftrightarrow \mathbf{x}H^t = \mathbf{y}H^t \Leftrightarrow (\mathbf{x} - \mathbf{y})H^t = 0 \Leftrightarrow \mathbf{x} - \mathbf{y} \in \mathcal{C} \Leftrightarrow \mathbf{x} + \mathcal{C} = \mathbf{y} + \mathcal{C}$

□

Theorem 4.7. *There is a 1-1 correspondence between syndromes and cosets.*

Proof. By 3) in Proposition 4.6 we have that all words in a coset have the same syndrome and that all words with the same syndrome belong to the same coset. □

Taking this into account, for any $[n,k]_q$-linear code $\mathcal{C}$ over $\mathbb{F}_q^n$, a smaller table can be constructed, so a more efficient decoding method is obtained. The *syndrome decoding method* works as the standard array method but instead of listing all q^n words in $\mathbb{F}_q^n$ the table used has only $2q^{n-k}$ entries that are the q^{n-k} coset leaders for the code and its syndromes. So whenever a word is received its syndrome has to be computed so if it is $\mathbf{0}$ it is assumed that no noise has occurred during the transmission process and no correction is needed but if it is not null, its coset leader is taken as the error vector and the word is corrected as $\mathbf{v} = \mathbf{y} - \mathbf{e}$ which corresponds to the nearest codeword for the received word as it happened when using the standard array.

EXAMPLE 4.3. Using the same code as in Example 4.1 we want to decode the same words as in Example 4.2. As the generator matrix G is systematic using Proposition 3.11, its systematic check matrix is,

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

and the array containing each coset leader with the syndrome of all the words that belong to the coset is,

coset leader	syndrome
0000	00
1000	21
0100	22
0010	10
0001	01
2000	12
0200	11
0020	20
0002	02

so now in order to decode the received words the first step is to calculate its syndrome and use the table as follows:

- If $\mathbf{y} = 1211$ is received, its syndrome is $s(\mathbf{y}) = \mathbf{y}H^T = 10$ so finding it in the table $\mathbf{e} = 0010$ and it is decoded as $\mathbf{v} = 1211 - 0010 = 1201$.
- If $\mathbf{y} = 2102$ is received, its syndrome is $s(\mathbf{y}) = \mathbf{y}H^T = 00$ so finding it in the table $\mathbf{e} = 0000$ and it is decoded as $\mathbf{v} = 2102 - 0000 = 2102$.

Note that in that case just 9 syndromes have to be stored. Furthermore, if we take into account that from the forth row on each codeword and syndrome is twice one of the already listed ones, so only 4 syndromes would need to be stored. □

Chapter 5
Maximum Distance Separable codes over fields

As has been explained in Chapter 1 the central problem of coding theory is to fix q, n and δ and find the greatest value of M (denoted by $\mathcal{A}_q(n,\delta)$) such that an $(n, M, \delta)_q$-code exists. Now, recall that in an $[n,k]_q$-linear-code the maximum number of codewords is q^k so the Singleton bound (Theorem 1.17) in now equivalent to:

$$q^k \leq q^{n-\delta+1}$$

and therefore it implies the following theorem.

THEOREM 5.1. *(Singleton bound for linear codes) Let $\mathcal{C}$ an $[n,k]_q$-linear-code. Then the Singleton bound states,*

$$k \leq n - \delta + 1.$$

This way in which the Singleton bound can be written is the reason why when working with linear codes another problem may be set. The most common problem now is to fix n and k and look for the largest minimum distance δ among all codes of length n and dimension k.

COROLLARY. *For an $[n,k]_q$-linear code we have $\delta \leq n - k + 1$.*

DEFINITION 5.2. A linear code which meets the Singleton bound for linear codes is called a *Maximum Distance Separable code* or *MDS-code*.

PROPOSITION 5.3. *An $[n,k]_q$-linear code is an MDS-code if and only if the minimum non-zero weight of any codeword is $n - k + 1$.*

PROOF. A linear code $\mathcal{C}$ is an MDS-code if and only if it satisfies the Singleton bound for linear codes (Theorem 5.1) $k = n - \delta + 1$ so if $\delta(\mathcal{C}) = n - k + 1$. Therefore as seen in Proposition 3.2 if and only if the minimum non-zero weight of any codeword is $n - k + 1$. □

THEOREM 5.4. *Let $\mathcal{C}$ be an $[n,k]_q$-linear code with a check matrix H. The minimum distance of $\mathcal{C}$, $\delta(\mathcal{C})$, is equal to the smallest positive number of columns of H which are linearly dependent.*

PROOF. Let the columns of H be designated as $H_1, ..., H_n$. Then since $\mathbf{u}H^T = 0$ for any codeword $\mathbf{u} = (u_1, ..., u_n) \in \mathcal{C}$, we have

$$u_1H_1 + \cdots + u_nH_n = \mathbf{u}H^T = \mathbf{0}.$$

Therefore, the nonzero coordinates of $\mathbf{u}$ give a non trivial linear combination of the columns of H. Thus, there is a $\mathbf{u} = (u_1, ..., u_n) \in \mathcal{C}$ whose weight is r if and only if r columns of H linearly dependent exist. The minimum distance of a linear code is the minimum non-zero weight of any codeword. So, it is also the minimum number of columns of H that are linearly dependent. □

THEOREM 5.5. *Let $\mathcal{C}$ be an $[n, k]_q$-linear code. Let G and H be a generator matrix and a check matrix, respectively, for $\mathcal{C}$. Then, the following statements are equivalent,*

(1) $\mathcal{C}$ is an MDS-code; .
(2) every set of $n - k$ columns of H is linearly independent;
(3) every set of k columns of G is linearly independent;
(4) $\mathcal{C}^\perp$ is an MDS-code;

PROOF. (1) $\Leftrightarrow$ (2) A code $\mathcal{C}$ is an MDS-code if and only if $\delta(\mathcal{C}) = n - k + 1$. By Theorem 5.5 we have that $\delta(\mathcal{C})$ is the smallest positive number of columns of H which are linearly dependent. Therefore $\mathcal{C}$ is MDS if and only if every set of $n - k$ columns of H is linearly independent.

(1) $\Rightarrow$ (4) Recall that if H is a check matrix for $\mathcal{C}$ then it is a generator matrix for $\mathcal{C}^\perp$, so the length of $\mathcal{C}^\perp$ is n and the dimension is $n-k$. To show that $\mathcal{C}^\perp$ is MDS, we need to show that its minimum distance $\delta(\mathcal{C}^\perp)$ is equal to $\delta' = n-(n-k)+1 = k+1$.

Suppose $\delta' \leq k$. Then there exist a codeword $\mathbf{u} \in \mathcal{C}^\perp$ with at most k non-zero coordinates and hence, at least $n-k$ zero coordinates. As permuting the coordinates of a word does not change its weight, it can be assumed that the last $n-k$ coordinates of $\mathbf{u}$ are 0. Considering H as $H = (A|H')$, where A is some $(n-k) \times k$ matrix and H' is a square $(n-k) \times (n-k)$ matrix. We know that the $n-k$ columns of H' are linearly independent because we have proved that (1) and (2) are equivalent so H' is invertible. Hence, the rows of H' are linearly independent. This means that the only word that can be encoded using H in order to obtain 0 in all the last $n-k$ coordinates, such as for $\mathbf{u}$, is the word $\mathbf{0}$. If not we would have a contradiction since we would have a linear combination of the linearly independent rows of H' that is equal to 0. Therefore, the entry codeword for obtaining $\mathbf{u}$ is the all-zero word $\mathbf{0}$. Consequently, $\delta' \geq k + 1$. The Singleton Bound for linear codes tells us that $\delta' \leq n - (n - k) + 1 = k + 1$. Putting both together it follows that $\delta' = k + 1$ so $\mathcal{C}^\perp$ is also an MDS code.

Since $(\mathcal{C}^\perp)^\perp = \mathcal{C}$, the above also shows (4) $\Rightarrow$ (1) and since G is a check matrix for $(\mathcal{C}^\perp)$ analogously to the case (1) $\Leftrightarrow$ (2) a code $\mathcal{C}^\perp$ is an MDS-code if and only if $\delta(\mathcal{C}^\perp) = k + 1$. By Theorem 5.5 it is known that $\delta(\mathcal{C}^\perp)$ is the smallest positive number of columns of G which are linearly dependent. Therefore $\mathcal{C}^\perp$ is MDS if and only if every set of k columns of G is linearly independent which proves (4) $\Leftrightarrow$ (3). □

COROLLARY. *Let H be a check matrix for an $[n,k]_q$-linear code $\mathcal{C}$. The code $\mathcal{C}$ is an MDS code if and only if every $h \times h$ subdeterminant of H is non-zero in $\mathbb{F}_q$ where $h = 1, ..., min\{n-k, k\}$.*

The importance of MDS-codes lies in the fact that they are of great interest because for a given value of n and k they are the codes with greater error detection and correction capability. Therefore, for a given value of k and knowing the number of elements q in the alphabet, it is natural to wonder which are the parameters of the best MDS linear code that can be constructed. As consequence of Lemma 1.9 a code is better when its minimum distance is as large as possible. Hence, as we are working with MDS linear codes, maximizing $\delta(\mathcal{C})$ is the same as maximizing n because $\delta = n-k+1 \Rightarrow n = \delta+k-1$. And that leads us to the following theorems that for a given k give some bounds for n so that an MDS linear code exists.

Let us first point out that the length of any MDS linear code $\mathcal{C}$ of dimension k is greater or equal than k. As otherwise we would have $\delta(\mathcal{C}) = n-k+1 < k-k+1 = 1$ which cannot happen.

THEOREM 5.6. *There exists an MDS linear code of dimension k and length $k+1$ over any $\mathbb{F}_q$ for any $k \geq 1$.*

PROOF. For any given k the following matrix

$$G = \left(\begin{array}{c|c} & 1 \\ I_k & \vdots \\ & 1 \end{array} \right)$$

where I_k is the $k \times k$ identity matrix, generates an MDS code of length $k+1$.

It is clear that it is a linear code of length $k+1$ (its rows are linearly independent and are basis of a subspace of $\mathbb{F}_q^{k+1}$) which encodes any word $\mathbf{x} = x_1...x_k \in \mathbb{F}_q^k$ as $\mathbf{u} = u_1...u_{k+1} \in \mathbb{F}_q^{k+1}$ where $u_i = x_i$ for $1 \leq i \leq k$ and $u_{k+1} = \sum_{i=1}^{k} x_i$. In order to see that it is an MDS we need to verify that $\delta(\mathcal{C}) = n-k+1$ thus that $\delta = (k+1)-k+1 = 2$. By Proposition 3.11 its check matrix is

$$H = (-1 \cdots -1 \mid 1)$$

so a $1 \times (k+1)$ with the first k positions equal to -1. Therefore, by Theorem 5.4, $\delta(\mathcal{C}) = 2$ for any $k \geq 1$. □

THEOREM 5.7. *Let $\mathcal{C}$ be an MDS linear code of dimension k over $\mathbb{F}_q$. If $k \geq q$,*

$$n \leq k+1.$$

PROOF. As we have just proved in Theorem 5.6 there exist an $[k+1, k]_q$ MDS linear code for all k. Let us suppose G to be a generator matrix of an $[m,k]_q$-linear code $\mathcal{C}$ with $m \geq k+2$ and see that then $\mathcal{C}$ is not MDS.

In Chapter 3 we have seen that any generator matrix of a linear code over a field can be converted into the form

$$G = \left(\begin{array}{ccc|ccc} 1 & \cdots & 0 & | & & | \\ \vdots & \ddots & \vdots & P_{k+1} & \cdots & P_m \\ 0 & \cdots & 1 & | & & | \end{array}\right)$$

where the first k columns from I_k and P_j with $k+1 \leq j \leq m$ are $k \times 1$ vectors which form a matrix P. Then all entries in each P_j must be non-zero. That is because if there was a P_j with a zero entry in the i-th row, the subdeterminant of G formed by all the columns of I_k, except the i-th column, and P_j would be equal to zero so by Theorem 5.5 $\mathcal{C}$ would not be MDS. Then every element on the column P_{k+1} of G has an inverse and by multiplying each i-th row of G by the inverse of the element in position i, $k+1$ we obtain an equivalent generator matrix of form

$$G' = \left(\begin{array}{ccc|ccc} \lambda_1 & \cdots & 0 & 1 & & | \\ \vdots & \ddots & \vdots & P'_{k+1} & \cdots & P'_m \\ 0 & \cdots & \lambda_k & 1 & & | \end{array}\right).$$

By the Pigeonhole principle, as in $\mathbb{F}_q$ there are only $q-1$ non-zero elements and each P'_j column has $k \leq q$ entries, there must be at least two equal elements in each column of P. Thus, assuming without lost of generality that in column P_{k+2} these two entries are the first and the second element of the vector,

$$\widetilde{G'} = \left(\begin{array}{ccc:cc} 0 & \cdots & 0 & 1 & \lambda \\ 0 & \cdots & 0 & 1 & \lambda \\ \hdashline & & & 0 & 0 \\ & I_{k-2} & & \vdots & \vdots \\ & & & 0 & 0 \end{array}\right)$$

is a submatrix of G' which determinant would be equal to 0 and again by Theorem 5.5 $\mathcal{C}$ would not be MDS. As this will happen for any P_j taking the appropriate columns of I_k we conclude that there is no $[m,k,3]_q$ MDS linear code for $k \leq q$ if $m \geq k+2$. Therefore, as we wanted to see, if $\mathcal{C}$ is an MDS linear code of dimension k over $\mathbb{F}_q$ and $k \geq q$ then n must be less or equal to $k+1$. □

THEOREM 5.8. *Let $\mathcal{C}$ be an MDS linear code of dimension k over $\mathbb{F}_q$. If $k \leq q$,*

$$n \leq q + k - 1.$$

PROOF. Let $U = \{u_1, \ldots, u_{k-2}\} \subseteq \mathbb{F}_q^k$ be a set of $k-2$ linearly independent vectors in $\mathbb{F}_q^k$, hence $\dim(\langle U \rangle) = k-2$. Now consider the set $\mathcal{V} \subset \mathbb{F}_q^k$, defined as

$$\mathcal{V} = \left\{ v \in \mathbb{F}_q^k \mid \dim(\langle U, v \rangle) = k-1 \right\}.$$

The cardinality of $\mathcal{V}$ is $|\mathcal{V}| = q^k - q^{k-2}$, because we are taking all vector in $\mathbb{F}_q^k$ and removing all the ones generated by U. As there are different vectors of $\mathcal{V}$ that generate the same subspace, we define the following equivalence relation in $\mathcal{V}$

$$v \sim w \;\Leftrightarrow\; \langle U, v \rangle = \langle U, w \rangle.$$

There are $q^{k-1} - q^{k-2}$ elements in each equivalence class because there q^{k-1} different vectors in a hyperplane of the form $\langle U, v \rangle$ but q^{k-2} are already contained in $\langle U \rangle$.

Therefore each of the remaining $q^{k-1} - q^{k-2}$ elements together with U generate the same subspace of dimension $k-1$.

Then we know that,

$$|\mathcal{V}/\sim| = \frac{q^k - q^{k-2}}{q^{k-1} - q^{k-2}} = \frac{q^{k-2}(q^2-1)}{q^{k-2}(q-1)} = q+1.$$

In that way, we know that if we construct the following matrix,

$$G = \left(\begin{array}{cccc|cccc} | & | & & | & | & | & & | \\ u_1 & u_2 & \cdots & u_{k-2} & v_1 & v_2 & \cdots & v_{q+1} \\ | & | & & | & | & | & & | \end{array}\right)$$

where $v_1, \ldots, v_{q+1}$ are representatives of each equivalence class, each $k \times k$ determinant of the form $det(u_1, \ldots, u_{k-2}, v_i, v_j)$ with $1 \leq i, j \leq q+1$ and $i \neq j$ will not be null.

By Theorem 5.5 we know that an $[n,k]_q$-linear code $\mathcal{C}$ is MDS if and only if every set of k columns of G is linearly independent. Therefore any generator matrix of an MDS linear code of dimension k has up to $(k-2)+(q+1)$ columns. So, as we wanted to see $n \leq q+k-1$. □

In the case that q is a prime there exists a better bound. The following theorem from [**1**].

THEOREM 5.9. *Let $\mathcal{C}$ be an MDS linear code of dimension k over $\mathbb{F}_q$. If $k \leq q$ and q is prime,*

$$n \leq q+1.$$

The last three theorems can be rewritten as,

THEOREM 5.10. *Let $\mathcal{C}$ be an MDS linear code of dimension k over $\mathbb{F}_q$.*

(1) If $k \leq q$ and
  (a) q is prime, then $n \leq q+1$.
  (b) q is not prime, then $n \leq q+k-1$
(2) If $k \geq q$, $n \leq k+1$.

The aim of the next Chapter is to study Reed Solomon codes which are linear MDS codes which meet this bound. Moreover, they are almost the only codes meeting this bound.

The following is also from [**1**].

THEOREM 5.11. *If q is prime and $k \leq q$ and $k \neq \frac{q+1}{2}$, then a linear MDS code of length $q+1$ is linearly equivalent to a Reed-Solomon code.*

Chapter 6
Reed-Solomon codes

In this chapter we construct linear k-dimensional MDS codes of length $q+1$ over $\mathbb{F}_q$ for all $k \leq q-1$. So they are an example of codes achieving the bound in Theorem 5.10.

Let $\mathbb{F}_q = \{a_0, a_2, .., a_{q-1}\}$ be a finite field and k an integer. Note that every word $\mathbf{b} = (b_0, b_1, ..., b_{k-1}) \in \mathbb{F}_q^k$ can be identified with a polynomial

$$f = b_0 + b_1 x + \cdots + b_{k-1} x^{k-1} \in \mathbb{F}_q[x] \tag{6.1}$$

so that b_i is the coefficient of x^i in that polynomial f which will be denoted by $coef_f(x^i)$.

Then, a *Reed-Solomon code* of length n for codifying words in $\mathbb{F}_q^k$ with $k \leq q-1$ is denoted $\mathcal{RS}(n,k)$ and can be constructed as follows:

Each word $\mathbf{b} = (b_0, b_1, ..., b_{k-1}) \in \mathbb{F}_q^k$, or equivalently its polynomial f is codified as a vector in $\mathbb{F}_q^{q+1}$ that has as its first q coordinates the word's corresponding polynomial evaluated in each of the q elements of the field and in its last one the the coefficient of x^{k-1} (the last letter of the word that is being encoded):

$$\mathbf{u} = (f(a_0),\ \ldots, f(a_{q-1}), coef_f(x^{k-1})).$$

It is clear that $n = q+1$ so given a field $\mathbb{F}_q$ and an integer $k \leq q+1$ the corresponding Reed-Solomon code $\mathcal{C} = \mathcal{RS}(n,k)$ is:

$$\mathcal{RS}(n,k) = \{(f(a_0), \ldots, f(a_{q-1}), coef_f(x^{k-1})) \mid f \in \mathbb{F}_q[x],\ deg(f) \leq k-1\}.$$

PROPOSITION 6.1. *Reed-Solomon codes, $\mathcal{RS}(n,k)$, are $[n,k]_q$-linear codes and their size is $M = q^k$.*

PROOF. Let $\mathcal{RS}(n,k)$ be a Reed-Solomon code, in order to be an $[n,k]_q$-linear code it must be shown, by Definition 3.1, that it is a linear subspace of dimension k of $\mathbb{F}_q^n$. Therefore it satisfies all conditions in Definition 2.4. By construction all codewords belong to $\mathbb{F}_q^n$, hence it is clear that $\mathcal{RS}(n,k) \subseteq \mathbb{F}_q^n$. Now, let $\mathbf{u} = (f(a_0),\ \ldots, f(a_{q-1}), coef_f(x^{k-1}))$ and $\mathbf{v} = (g(a_0),\ \ldots, g(a_{q-1}), coef_g(x^{k-1}))$ two different codewords for some polynomials $f, g \in \mathbb{F}_q[x]$ with degree less or equal than $k-1$, and some $\lambda \in \mathbb{F}_q$, then $f+g$ and λf are also polynomials in $\mathbb{F}_q[x]$ with degree

less or equal than $k-1$ so:

$$\mathbf{u}+\mathbf{v}=(f(a_0)+g(a_0),\ \ldots,f(a_{q-1})+g(a_{q-1}),coef_f(x^{k-1})+coef_g(x^{k-1}))$$

and

$$\lambda\mathbf{u}=(\lambda f(a_0),\ \ldots,\lambda f(a_{q-1}),\lambda coef_f(x^{k-1}))$$

also belong to $\mathcal{RS}(n,k)$. Thus $\mathcal{RS}(n,k)$ is an $[n,k]_q$-linear code and as the size of a linear code is q^k,

$$|\mathcal{RS}(n,k)|=q^k$$

which already clear by construction since there are q^k such polynomials $f\in\mathbb{F}_q[x]$ of $deg(f)\leq k-1$ (there are q choices for each coefficient a_i).

□

THEOREM 6.2. *Reed-Solomon, $\mathcal{RS}(n,k)$, codes are MDS-codes.*

PROOF. In order to be an MDS-code we want to prove that $\delta=n-k+1$. Let $\mathbf{u}$ be a codeword generated using f and note that by construction the degree of the polynomial f cannot be greater than $k-1$ so:

- if $deg(f)=k-1$, we know that the last coordinate of $\mathbf{u}$ is different from 0. As a polynomial of degree $k-1$ has at most $k-1$ roots, $w(\mathbf{u})\geq n-(k-1)$. This is true for any codeword hence $\delta=w(\mathcal{C})\geq n-k+1$. By the Singleton bound for linear codes $\delta\leq n-k+1$ so $\delta=n-k+1$ and $\mathcal{RS}(n,k)$ is MDS-codes.
- if $deg(f)\leq k-2$, we know that the last coordinate of $\mathbf{u}$ is equal to 0 and as polynomial of degree $k-2$ has at most $k-2$ roots $w(\mathbf{u})\geq n-(k-2)-1$. This is true for any codeword hence $\delta=w(\mathcal{C})\geq n-k+1$. By Singleton bound for linear codes $\delta\leq n-k+1$ so $\delta=n-k+1$ and $\mathcal{RS}(n,k)$ is MDS-codes.

□

COROLLARY. *$\mathcal{RS}(n,k)$ codes are $[n,k,\delta]_q$-linear codes with $\delta=n-k+1$.*

As linear codes, $\mathcal{RS}(n,k)$ codes can also be defined as

$$\mathcal{C}=\{\mathbf{u}G|\mathbf{u}\in\mathbb{F}_q^k\}$$

where G is the generator matrix of the code.

LEMMA 6.3. *Let $\mathcal{RS}(n,k)$ then a generator matrix $G\in\mathbb{F}_q^{n\times k}$ is given by*

$$G=\begin{pmatrix}1 & 1 & 1 & \cdots & 1 & 0\\ a_0 & a_1 & a_2 & \cdots & a_{q-1} & 0\\ a_0^2 & a_1^2 & a_2^2 & \cdots & a_{q-1}^2 & 0\\ \vdots & \vdots & \vdots & & \vdots & \vdots\\ a_0^{k-2} & a_1^{k-2} & a_2^{k-2} & \cdots & a_{q-1}^{k-2} & 0\\ a_0^{k-1} & a_1^{k-1} & a_2^{k-1} & \cdots & a_{q-1}^{k-1} & 1\end{pmatrix}.$$

where $a_0,...,a_{q-1}$ denote the q elements of $\mathbb{F}_q$.

PROPOSITION 6.4. *Let H be the following matrix*

$$H = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 0 \\ a_0 & a_1 & a_2 & \cdots & a_{q-1} & 0 \\ a_0^2 & a_1^2 & a_2^2 & \cdots & a_{q-1}^2 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_0^{n-k-2} & a_1^{n-k-2} & a_2^{n-k-2} & \cdots & a_{q-1}^{n-k-2} & 0 \\ a_0^{n-k-1} & a_1^{n-k-1} & a_2^{k-1} & \cdots & a_{q-1}^{n.k-1} & 1 \end{pmatrix} \tag{6.2}$$

in $\mathbb{F}_q^{n\times n-k}$ where $a_0, ..., a_{q-1}$ denote the q elements of $\mathbb{F}_q$. H is a check matrix for $\mathcal{RS}(n,k)$.

PROOF. By Lemma 3.7 we need to see that the rank of H is $n-k$ and that $GH^T = \mathbf{0}$.

The rank of H is $n-k$ as the determinant of the $n-k$ first columns is nonsingular as it is a *Vandermonde matrix* where all a_i are distinct.

$$GH^T = \begin{pmatrix} q & \sum_{i=0}^{q-1} a_i & \sum_{i=0}^{q-1} a_i^2 & \cdots & \sum_{i=0}^{q-1} a_i^{n-k-1} \\ \sum_{i=0}^{q-1} a_i & \sum_{i=0}^{q-1} a_i^2 & \sum_{i=0}^{q-1} a_i^3 & \cdots & \sum_{i=0}^{q-1} a_i^{n-k} \\ \sum_{i=0}^{q-1} a_i^2 & \sum_{i=0}^{q-1} a_i^3 & \sum_{i=0}^{q-1} a_i^4 & \cdots & \sum_{i=0}^{q-1} a_i^{n-k+1} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum_{i=0}^{q-1} a_i^{k-2} & \sum_{i=0}^{q-1} a_i^{k-1} & \sum_{i=0}^{q-1} a_i^k & \cdots & \sum_{i=0}^{q-1} a_i^{n-3} \\ \sum_{i=0}^{q-1} a_i^{k-1} & \sum_{i=0}^{q-1} a_i^k & \sum_{i=0}^{q-1} a_i^{k+1} & \cdots & \sum_{i=0}^{q-1} a_i^{n-2} + 1 \end{pmatrix}$$

all the summations are equal to 0 by Lemma 2.19 but the one in the lower right corner that as $n = q+1$ is equal to -1 so when added 1 is also 0. □

THEOREM 6.5. *The dual code of a Reed Solomon code $\mathcal{RS}(n,k)$*

$$\mathcal{C} = \{(f(a_0), \ldots, f(a_{q-1}), coef_f(x^{k-1})) \mid f \in \mathbb{F}_q[x],\ deg(f) \leq k-1\}$$

is a also a Reed Solomon code, $\mathcal{RS}(n, n-k)$

$$\mathcal{C}^\perp = \{(g(a_0), \ldots, g(a_{q-1}), coef_g(x^{k-1})) \mid g \in \mathbb{F}_q[x],\ deg(g) \leq n-k-1\}.$$

PROOF. By construction 6.2 is the generator matrix of and $\mathcal{RS}(n, n-k)$. It has been proved previously that check matrix of a code is the generator matrix of it's dual. Therefore $\mathcal{RS}(n,k)^\perp = \mathcal{RS}(n, n-k)$. □

COROLLARY. *If $k \leq \frac{q+1}{2}$ a $\mathcal{RS}(n,k)$ code is self-orthogonal and moreover, if $k = \frac{q+1}{2}$ then it is self-dual.*

EXAMPLE 6.1. We want to construct an $\mathcal{RS}(4,2)$ code.

As $n = q+1$ $q = 3$ and we are working in $\mathbb{F}_3 = \{0,1,2\}$. The size of the code is $3^2 = 9$.

Let us construct in both ways by definition and by construction it's generator matrix

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

in the following table $\mathbf{x} \in \mathbb{F}_3^2$ denotes the word that has to be encoded, f in form 6.1 that $\mathbf{x}$ determines and $\mathbf{u} \in \mathbb{F}_3^4$ the codeword.

$\mathbf{x}$	f	$\mathbf{u} = ($	$f(0),$	$f(1),$	$f(2),$	$coef_f(x))$	$\mathbf{u} = \mathbf{x}G$
00	0		0	0	0	0	0000
01	x		0	1	2	1	0121
10	1		1	1	1	0	1110
11	$1+x$		1	2	0	1	1201
02	$2x$		0	2	1	2	0212
20	2		2	2	2	0	2220
12	$1+2x$		1	0	2	2	1022
21	$2+x$		2	0	1	1	2011
22	$2+2x$		2	1	0	2	2102

It's check matrix is

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

so $GH^T = \mathbf{0}$ and $\mathcal{RS}^T(4,2) = \mathcal{RS}(4,2)$, it is a self-dual code. □

Chapter 7
Linear codes over commutative rings

In this chapter, let R be a commutative ring with identity.

DEFINITION 7.1. Let A be an $n \times n$ matrix over R and let A_{ij} denote the $(n-1) \times (n-1)$ matrix obtained from A by deleting the $i-th$ row and the $j-th$ column. Set $b_{ij} = (-1)^{i+j} det(A_{ij})$ so $B = [b_{ij}]$. Then *the adjoint of* A is $Adj(A) = B^T$.

The following is from [**6**].

THEOREM 7.2. *Let A be an square matrix then*

- $A \cdot Adj(A) = det(A) \cdot I = Adj(A) \cdot A$.
- *if A is invertible then $det(A)$ is a unit and $A^{-1} = (det(A))^{-1} \cdot Adj(A)$.*

DEFINITION 7.3. An element x in a commutative ring R is called *annihilator* of an element $y \neq 0$ in R-module N if $xy = 0$. If $N = R$, then an annihilator is called *zero divisor* of R.

DEFINITION 7.4. A *linear code of length n over R* is an R-submodule of R^n.

DEFINITION 7.5. A *free linear code of length n over R* is a free R-submodule of R^n.

PROPOSITION 7.6. *In a linear code of length n over R the minimum distance is equal to the minimum weight among all non-zero codewords. In other words,*

$$\delta(\mathcal{C}) = \min\{w(\mathbf{u}) \mid \mathbf{0} \neq \mathbf{u} \in \mathcal{C}\}$$

PROOF. The proof given in Proposition 3.2 holds over rings. □

DEFINITION 7.7. A matrix G is called a *generator matrix* for a linear code $\mathcal{C}$ over R if its rows span $\mathcal{C}$ and none of them can be written as a linear combination of the other rows. That is, if G is a $k \times n$ matrix such that

$$\mathcal{C} = \{(u_1, u_2, ..., u_k)G | u_i \in R^k\}.$$

Note that unlike what happened in Chapter 3, now the rows $\mathbf{r}_1, ..., \mathbf{r}_k$ of G should be regarded as elements of R^n not $\mathbb{F}^n$,so not all the elementary row operations listed there can be used. In order to not change the submodule $\mathcal{C}$ spanned by the rows of G the following row operations can be used.

LEMMA 7.8. *(Elementary row operations over rings)*

- ***Row switching:*** *($\mathbf{r}_i \leftrightarrow \mathbf{r}_j$)*
  A row within the matrix can be switched with another row.
- ***Row multiplication:*** *($k\mathbf{r}_i \to \mathbf{r}_i, k \in \mathcal{U}(R)$)*
  A row can only be multiplied by a unit .
- ***Row addition:*** *($\mathbf{r}_i + k\mathbf{r}_j \to \mathbf{r}_i, i \neq j$)*
  A row can be replaced by the sum of that row and a multiple of another row.

From now on we will restrict ourselves to the case where R is a p-adic ring, hence $R = \mathbb{Z}/p^l\mathbb{Z}$.

DEFINITION 7.9. A generator matrix G of a linear code over a p-adic ring R is a *standard generator matrix* if it is has the form

$$G = \begin{pmatrix} I_{k_0} & A_{0,1} & A_{0,2} & A_{0,3} & \cdots & A_{0,l-1} & A_{0,l} \\ 0 & pI_{k_1} & pA_{1,2} & pA_{1,3} & \cdots & pA_{1,l-1} & pA_{1,l} \\ 0 & 0 & p^2I_{k_2} & p^2A_{2,3} & \cdots & p^2A_{2,l-1} & p^2A_{2,l} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & p^{l-1}I_{k_{l-1}} & p^{l-1}A_{l-1,l} \end{pmatrix} \tag{7.1}$$

where the columns are grouped into blocks of sizes $k_0,\ k_1,\ ..., k_{l-1}, k_l$ where $k_l = n - \sum_{i=0}^{l-1} k_i$ with $k_i \geq 0$ for $0 \leq i \leq l-1$ thus, I_{k_i} denotes the $k_i \times k_i$ identity matrix for any i.

THEOREM 7.10. *Every generator matrix G of a linear code over a ring R can be expressed after some column permutations or rows elementary operations (described in Lemma 7.8), as a standard generator matrix.*

PROOF. Every element $a \in \mathbb{Z}/p^l\mathbb{Z}$ can be written uniquely as a finite sum

$$a = a_0 + a_1p^1 + a_2p^2 + \cdots + a_{l-1}p^{l-1} \tag{7.2}$$

where $0 \leq a_i \leq p-1$. Note that the units in $\mathbb{Z}/p^l\mathbb{Z}$ are then

$$\mathcal{U}(\mathbb{Z}/p^l\mathbb{Z}) = \{u \in \mathbb{Z}/p^l\mathbb{Z} \mid u_0 \neq 0 \text{ when expressed in form (7.2)}\}.$$

Therefore, given any $k \times n$ generator matrix

$$G = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{k,1} & a_{k,2} & \cdots & a_{k,n} \end{pmatrix}.$$

rearranging its rows and columns by rows and columns permutations it is possible to have a $k_0 \times k_0$ square submatrix A'_{k_0} in the upper left corner so that all elements $a_{i,i}$ in it have $a_0 \neq 0$ when expressed in form (7.2). Therefore, as they are units they all have an inverse so it is possible to make all matrix elements below A'_{k_0} vanish and A'_{k_0} to be the $k_0 \times k_0$ identity I_{k_0}. It is clear that there is no element $a_{i,j}$ below the k_0-row with $a_0 \neq 0$. Then, analogously, the same procedure can be done but instead of the moving all elements whose expression in (7.2) form have $a_o \neq 0$ we pick the ones with $a_1 \neq 0$. A $k_1 \times k_1$ submatrix A'_{k_1} with all elements $a_{i,i}$ multiples of p can be constructed just below and after the first k_0 rows and

columns. Again, it is possible to make all matrix elements below A'_{k_1} vanish and A'_{k_0} to be the p times the $k_1 \times k_1$ identity I_{k_1}. It is clear that there is no element $a_{i,j}$ below the k_1-row with a_0 and a_1 different from 0. By repeating this procedure until $a_{l-1} \neq 0$ a generator matrix in form (7.1) is obtained. Note that some k_i might be equal to 0. □

DEFINITION 7.11. It is said that a linear code $\mathcal{C}$ over R has *type*,

$$(k_0, k_1, .., k_{l-1})$$

where k_i are the corresponding sizes of its generator matrix in standard form. The zero code (containing only the zero codeword) has type 0.

REMARK 7.1. Recall that a linear code $\mathcal{C}$ is a free code, if it is a free R-submodule of R^n. This implies that if $\mathcal{C}$ is a free code, the rows of any of his generator matrices G form a basis of the free submodule. Therefore, a generator matrix of a free code G is a standard generator matrix if it has form,

$$G = (I_k \mid P) = \begin{pmatrix} 1 & & & & * & * & \cdots & * \\ & 1 & & & * & * & \cdots & * \\ & & \ddots & & \vdots & \vdots & & \vdots \\ & & & 1 & * & * & \cdots & * \end{pmatrix}.$$

Which means that free codes always admit a systematic generator matrix as in Chapter 3 when working over fields. Free codes over $\mathbb{Z}/p^l\mathbb{Z}$ are all of type $(k, 0, .., 0)$.

In Chapter 3 we saw that the size of an $[n, k]_q$-linear code is q^k. But, as can be seen in the following example, that does not hold when working over rings.

EXAMPLE 7.1. Consider the linear code of type $(1, 1)$ over $\mathbb{Z}/4\mathbb{Z}$ generated by

$$G = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & 2 \end{pmatrix}.$$

Therefore, as

$$(a, b)\begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & 2 \end{pmatrix} = (a, 2b, 3a + 2b)$$

the words $\mathbf{x} \in \mathbb{Z}/4\mathbb{Z}$ are coded as

$\mathbf{x} \in \mathbb{Z}/4\mathbb{Z}$	$\mathbf{x}G$	$\mathbf{x} \in \mathbb{Z}/4\mathbb{Z}$	$\mathbf{x}G$
00	000	20	202
01	022	21	220
02	000	22	200
03	022	23	220
10	103	30	301
11	121	31	323
12	103	32	301
13	121	33	323

Then our code is exactly

$$\mathcal{C} = \{(0,0,0), (0,2,2), (1,0,3), (1,2,1), (2,2,0), (2,0,0), (3,0,1), (3,2,3)\}$$

and as can be seen $M = \mid \mathcal{C} \mid = 8 \neq 4^2$.

Note that $\mathcal{C} = \{\mathbf{u} \in \mathbb{Z}/4\mathbb{Z}^3 \mid \mathbf{u} = \mathbf{x}G \ \forall \mathbf{x} \in \mathbb{Z}/4\mathbb{Z}^2\}$ is not a free code as the rows of G are not a basis. □

LEMMA 7.12. *The size of a p-adic code is $p^{k'}$, where*

$$k' = \sum_{i=0}^{l-1}(l-i)k_i.$$

REMARK 7.2. In a free code over R, $|\mathcal{C}| = p^{lk}$ as it happened when working with linear codes over fields.

DEFINITION 7.13. Let $\mathcal{C}$ be a p-adic code with generator matrix in form (7.1). Then $\mathcal{C}$ has a dual code $\mathcal{C}^{\perp}$ with generator matrix of the form

$$H = \begin{pmatrix} B_{0,l} & B_{0,l-1} & \cdots & B_{0,3} & B_{0,2} & B_{0,1} & I_{k_l} \\ pB_{1,l} & pB_{1,l-1} & \cdots & pB_{1,3} & pB_{1,2} & pI_{k_{l-1}} & 0 \\ p^2B_{2,l} & p^2B_{2,l-1} & \cdots & p^2B_{2,3} & p^2I_{k_{l-2}} & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ p^{l-1}B_{l-1,l} & p^{l-1}I_{k_1} & \cdots & 0 & 0 & 0 & 0 \end{pmatrix} \tag{7.3}$$

where the columns are grouped into blocks of the same sizes $k_0,\ k_1,\ ..., k_{l-1}, k_l$ as in Definition 7.9. Therefore, H is said to be a check matrix for $\mathcal{C}$ in standard form.

DEFINITION 7.14. It is said that the dual code $\mathcal{C}^{\perp}$ over R has *type*,

$$(k_l, k_{1-1}, .., k_1).$$

LEMMA 7.15. *The size of the dual code $\mathcal{C}^{\perp}$ is $p^{k'_{\perp}}$, where*

$$k'_{\perp} = \sum_{i=1}^{l} ik_i.$$

EXAMPLE 7.2. Consider the code $\mathcal{C}$, over the 2-adic ring $\mathbb{Z}/16\mathbb{Z}$, generated by

$$G = \begin{pmatrix} 2 & 5 & 6 & 1 & 0 & 8 & 6 & 0 \\ 4 & 2 & 11 & 10 & 8 & 0 & 6 & 0 \\ 2 & 10 & 10 & 13 & 8 & 8 & 14 & 0 \\ 0 & 2 & 3 & 4 & 0 & 8 & 2 & 0 \\ 6 & 14 & 0 & 7 & 0 & 0 & 0 & 8 \\ 2 & 4 & 15 & 0 & 0 & 0 & 4 & 8 \end{pmatrix}.$$

By applying the following elementary row operations and column permutations

$$\begin{pmatrix} 2 & 5 & 6 & 1 & 0 & 8 & 6 & 0 \\ 4 & 2 & 11 & 10 & 8 & 0 & 6 & 0 \\ 2 & 10 & 10 & 13 & 8 & 8 & 14 & 0 \\ 0 & 2 & 3 & 4 & 0 & 8 & 2 & 0 \\ 6 & 14 & 0 & 7 & 0 & 0 & 0 & 8 \\ 2 & 4 & 15 & 0 & 0 & 0 & 4 & 8 \end{pmatrix} \xrightarrow[\substack{c_1 \leftrightarrow c_4 \\ c_2 \leftrightarrow c_3 \\ c_5 \leftrightarrow c_7}]{} \left(\begin{array}{ccccc:c:ccc} 1 & 6 & 5 & 2 & 6 & 8 & 0 & 0 \\ 10 & 11 & 2 & 4 & 6 & 0 & 8 & 0 \\ 13 & 10 & 10 & 2 & 14 & 8 & 8 & 0 \\ 4 & 3 & 2 & 0 & 2 & 8 & 0 & 0 \\ 7 & 0 & 14 & 6 & 0 & 0 & 0 & 8 \\ 0 & 15 & 4 & 2 & 4 & 0 & 0 & 8 \end{array}\right)$$

$$\xrightarrow[\substack{r_2 \to r_2+6r_1 \\ r_3 \to r_3+3r_1 \\ r_4 \to r_4+12r_1 \\ r_5 \to r_5+9r_1}]{} \left(\begin{array}{ccccc:c:ccc} 1 & 6 & 5 & 2 & 6 & 8 & 0 & 0 \\ 0 & 15 & 10 & 0 & 10 & 0 & 8 & 0 \\ 0 & 12 & 6 & 8 & 0 & 0 & 8 & 0 \\ 0 & 11 & 2 & 8 & 10 & 8 & 0 & 0 \\ 0 & 6 & 2 & 8 & 6 & 8 & 0 & 8 \\ 0 & 15 & 4 & 2 & 4 & 0 & 0 & 8 \end{array}\right)$$

$$\xrightarrow[\substack{r_1 \to r_1+6r_2 \\ r_2 \leftrightarrow 15r_2 \\ r_3 \to r_3+12r_2 \\ r_4 \to r_4+11r_2}]{} \left(\begin{array}{ccccc:c:ccc} 1 & 0 & 1 & 2 & 2 & 8 & 0 & 0 \\ 0 & 1 & 6 & 0 & 6 & 0 & 8 & 0 \\ 0 & 0 & 14 & 8 & 8 & 0 & 8 & 0 \\ 0 & 0 & 4 & 8 & 8 & 8 & 8 & 0 \\ 0 & 6 & 2 & 8 & 6 & 8 & 0 & 8 \\ 0 & 15 & 4 & 2 & 4 & 0 & 0 & 8 \end{array}\right)$$

$$\xrightarrow[\substack{r_3 \to r_3+r_4 \\ r_5 \to r_5+10r_2 \\ r_6 \to r_6+r_2}]{} \left(\begin{array}{ccccc:c:ccc} 1 & 0 & 1 & 2 & 2 & 8 & 0 & 0 \\ 0 & 1 & 6 & 0 & 6 & 0 & 8 & 0 \\ \hdashline 0 & 0 & 2 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 4 & 8 & 8 & 8 & 8 & 0 \\ 0 & 0 & 14 & 8 & 2 & 8 & 0 & 8 \\ 0 & 0 & 10 & 2 & 10 & 0 & 8 & 8 \end{array}\right)$$

$$\xrightarrow[\substack{r_2 \to r_2+5r_3 \\ r_4 \to r_4+6r_3 \\ r_5 \to r_5+r_3 \\ r_6 \to r_6+3r_3}]{} \left(\begin{array}{ccccc:c:ccc} 1 & 0 & 1 & 2 & 2 & 8 & 0 & 0 \\ 0 & 1 & 0 & 0 & 6 & 8 & 8 & 0 \\ \hdashline 0 & 0 & 2 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 8 & 8 & 8 & 8 & 0 \\ 0 & 0 & 0 & 8 & 2 & 0 & 0 & 8 \\ 0 & 0 & 0 & 2 & 10 & 8 & 8 & 8 \end{array}\right)$$

$$\xrightarrow[\substack{r_4 \to r_4+5r_6 \\ r_5 \to r_5+r_4}]{} \left(\begin{array}{ccccc:c:ccc} 1 & 0 & 1 & 2 & 2 & 8 & 0 & 0 \\ 0 & 1 & 0 & 0 & 6 & 8 & 8 & 0 \\ \hdashline 0 & 0 & 2 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 2 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 8 & 8 & 8 \\ 0 & 0 & 0 & 2 & 10 & 8 & 8 & 8 \end{array}\right)$$

$$\xrightarrow[\substack{r_4 \to r_4+15r_5 \\ r_5 \to r_5+r_3 \\ r_5 \leftrightarrow 5r_5}]{} \left(\begin{array}{ccccc:c:ccc} 1 & 0 & 1 & 2 & 2 & 8 & 0 & 0 \\ 0 & 1 & 0 & 0 & 6 & 8 & 8 & 0 \\ \hdashline 0 & 0 & 2 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 8 & 8 & 0 \\ 0 & 0 & 0 & 0 & 2 & 8 & 8 & 8 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 8 & 8 & 0 \end{array}\right)$$

a standard generator matrix has been obtained.

Then, $\mathcal{C}$ has type $(2, 3, 0, 1)$ and $16^2 \cdot 8^3 \cdot 2$ elements. □

Chapter 8
Maximum Distance Separable codes over p-adic rings

In this chapter, we will now work on the characterization of MDS linear codes over p-adic rings.

THEOREM 8.1. *A free k dimensional linear code $\mathcal{C}$ with a check matrix of the form $H = (-M \mid I_{n-k})$ is MDS if and only if every $h \times h$ subdeterminant of M is not an annihilator in R where $h = 1, ..., min\{n-k, k\}$:*

PROOF. We first want to prove that if every $h \times h$ subdeterminant of M is not an annihilator in R then $\mathcal{C}$ is MDS. Thus what we want to prove that $\mathcal{C}$ meets the Singleton Bound for linear codes ($\delta(\mathcal{C}) = n - k + 1$), by Proposition 7.6 it is equivalent to show that $w(\mathcal{C}) = r + 1$, where $r = n - k$.

Assume $w(\mathcal{C}) \leq r$. Let then $\mathbf{u}$ be a codeword such that $w(\mathbf{u}) = r$, as $\mathbf{u} \in \mathcal{C}$ we know that $\mathbf{u}H^T = \mathbf{0}$. Therefore, we know that there exist $\lambda_1, \lambda_2, ..., \lambda_r \in R$, which are exactly the r nonzero coordinates of $\mathbf{u}$, that give a non trivial linear combination of columns of H so that

$$\lambda_1 H_{\lambda_1} + \cdots + \lambda_r H_{\lambda_r} = 0 \Rightarrow \begin{pmatrix} | & | & & | \\ H_{\lambda_1} & H_{\lambda_2} & \cdots & H_{\lambda_r} \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} = \mathbf{0}^T$$

where $H_{\lambda_1}, ..., H_{\lambda_r}$ are the r columns of H which are multiplied by each λ_i respectively. Since $H = (-M|I_r)$ we can suppose that h of the columns H_{λ_i} are columns of $-M$ and $r - h$ are from I_r. The entries of columns coming from I_r are all zeros but one 1, so let $\widetilde{M}$ be the matrix obtained by adding the columns H_{λ_i} that are from $-M$ and then deleting the rows which correspond to the position of the 1 of each of the H_{λ_i} that come from I_r. By construction $\widetilde{M}$ is an $h \times h$ square submatrix of $-M$. Let us rename the λ_i that multiply each of the h columns H_{λ_i} that come from $-M$ $\mu_1, \mu_2, ..., \mu_h$ respectively. So now we know that

$$\widetilde{M} \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_h \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Then using Theorem 7.2,

$$\widetilde{M}\mu^T = \mathbf{0}^T \quad \Rightarrow \quad Adj(\widetilde{M})\widetilde{M}\mu^T = \mathbf{0}^T \quad \Leftrightarrow \quad det(\widetilde{M})I_h\mu^T = \mathbf{0}^T.$$

By hypothesis $det(\widetilde{M})$ is a not an annihilator in R so we know $det(\widetilde{M})\mu_i = 0 \Leftrightarrow \mu_i = 0\ \forall i \Leftrightarrow \mu = \mathbf{0}$. And now recalling that

$$H\mathbf{u}^t = \mathbf{0} \Leftrightarrow (-M|I_r)\mathbf{u}^t = \mathbf{0}$$

we have found out that all the $n-k$ first entries of $\mathbf{u}$ are equal to 0. As we have supposed at the beginning that $w(\mathbf{u}) = r$ we must have then $\mathbf{u} = (0, ...0, \lambda_1, ..., \lambda_r)$ with $\lambda_1, ..., \lambda_r \neq 0$. But this can not happen, because then we would have

$$(-M|I_r)\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} = \mathbf{0}^T \Leftrightarrow \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

and that implies $\lambda_i = 0\ \forall i$. Therefore, $\mathbf{u}$ would be the word $\mathbf{0}$, whose weight is 0, that is a contradiction, thus $w(\mathcal{C}) > r$. As for linear codes $w(\mathcal{C}) = \delta(\mathcal{C})$ and Singleton Bound says $\delta(\mathcal{C}) \leq n - k + 1 = r + 1$ we can conclude that $w(\mathcal{C}) = r$.

On the other hand, we want to prove that if $\mathcal{C}$ is MDS then every $h \times h$ subdeterminant of M is not an annihilator:

Let $\widetilde{M}$ be an $h \times h$ submatrix of M. Let e be a maximal integer such that, p^e divides $det(\widetilde{M})$.

Assuming that $e \geq 1$, by Theorem 7.2 we know, $\widetilde{M}Adj(\widetilde{M}) = det(\widetilde{M})I_h$. So, multiplying both sides of the equality by p^{l-e} we obtain,

$$\widetilde{M}\left(p^{l-e}Adj(\widetilde{M})\right) = \mathbf{0}_h$$

where $\mathbf{0}_h$ is an all zero entries $h \times h$ matrix.

If $p^{l-e}Adj(\widetilde{M}) \neq 0$ then there is a column of $p^{l-e}Adj(\widetilde{M})$ which is non-zero. The elements of this column $\lambda_1, ..., \lambda_h$ gives us a linear combination of the columns of $\widetilde{M}$ which is zero. Let $\mathbf{u}$ be a word in R^n which has zero entries in all its coordinates that, in the product $H\mathbf{u}^t$, multiply the columns of M which are not in $\widetilde{M}$, and $\lambda_1, ..., \lambda_h$ in the corresponding coordinates so that in the product $H\mathbf{u}^t$ each column of $\widetilde{M}$ is multiplied by its corresponding λ_i of the mentioned linear combination. Then, the rest of the coordinates of $\mathbf{u}$ are zero but for $r-h$ of them that multiply certain columns of I_r ($r = n-k$). Assuming without lost of generality that $\widetilde{M}$ is in the upper left corner of M we would have, that they are the last $r-h$ coordinates. They have values $\mu_j = -\sum_{i=1}^{h} \lambda_i a_{j,i}$ for each $j \in \{h+1, h+2, ..., r-1, r\}$. Then

the product $H\mathbf{u}^t$

$$\left(\begin{array}{ccc|ccc||c} & & & a_{1,h+1} & \cdots & a_{1,k} & \\ & \widetilde{M} & & \vdots & & \vdots & \\ & & & a_{h,h+1} & \cdots & a_{h,k} & \\ \hline a_{h+1,1} & \cdots & a_{h+1,h} & a_{h+1,h+1} & \cdots & a_{h+1,k} & \\ \vdots & & \vdots & \vdots & & \vdots & \\ a_{r,1} & \cdots & a_{r,h} & a_{r,h+1} & \cdots & a_{r,k} & \end{array}\right. \quad \left. I_r \right) \left(\begin{array}{c} \lambda_1 \\ \vdots \\ \lambda_h \\ 0 \\ \vdots \\ 0 \\ \hline\hline 0 \\ \vdots \\ 0 \\ \mu_{h+1} \\ \mu_{h+2} \\ \vdots \\ \mu_r \end{array}\right),$$

is equal to $\mathbf{0}$. This means $\mathbf{u} \in \mathcal{C}$ as it satisfies the equations of the check matrix. Then the weight of $\mathbf{u}$ is $w(\mathbf{u}) \leq h + (r - h) = r = n - k$. So we have found a codeword whose weight equal or smaller than $n - k$, which is a contradiction because by hypothesis $\mathcal{C}$ is MDS so the weight of any codeword should be greater or equal to $n - k + 1$.

Therefore, $p^{l-e}Adj(\widetilde{M})$ must be equal to 0. That means that p^e divides every element of $Adj(\widetilde{M})$, so p^e divides $det(Adj(\widetilde{M}))$. Using again Theorem 7.2,

$$\widetilde{M}Adj(\widetilde{M}) = det(\widetilde{M})I_h \;\Rightarrow\; det(\widetilde{M})det(Adj(\widetilde{M})) = det(\widetilde{M}).$$

This is a contradiction, because the left hand side of the last equality has a p^{2e} factor while the right hand side has at most p^e by hypothesis.

Thus $e = 0$, $p \nmid det(\widetilde{M})$ so every $h \times h$ subdeterminant of M is not an annihilator, as we wanted to prove. □

Conclusion

We have proven that the bounds for MDS linear codes over $\mathbb{Z}/p\mathbb{Z}$ carry over to MDS codes over p-adic rings with check matrices of the form $H = (-M|I_{n-k})$.

It is possible that these bounds carry over to all p-adic MDS codes. It would be also interesting to study MDS codes over $\mathbb{Z}/n\mathbb{Z}$, where n is not a prime power. Primarily, we have proven that MDS codes over p-adic rings with check matrices of the form $H = (-M|I_{n-k})$ are always short; their length is at most $p + 1$. This could possibly be true for all p-adic MDS codes, and could be the basis of further investigation.

Notes

The basic results on the algebraic objects considered here, rings, fields, vector spaces and modules, can be found in [**4**]. The book by McWilliams and Sloane [**3**] is a basic text on Error-correcting codes, including chapters on MDS codes, on Reed Solomon codes and syndrome decoding. The book by Roman [**5**] has been used for the section on the main coding theory problem in Chapter 1 as well as for Chapter 5.

The article by Calderbank and Sloane [**2**] has been used as a reference for Chapter 7.

The book by Brunat and Ventura [**7**] has been useful for several parts of this text, for instance, for defining basic concepts about codes in Chapter 1, defining the dual of a Reed Solomon code in Chapter 6, for properties of linear codes over fields in Chapter 3 and to provide some properties about Galois rings in Chapter 2.

Finally, both [**9**] and [**10**] have been used for some definitions and results concerning modules, submodules and quotient rings.

References

[1] Simeon Ball, On sets of vectors of a finite vector space in which every subset of basis size is a basis, *J. Eur. Math. Soc. (JEMS)*, **14**, 733–748 (2012).

[2] A.R. Calderbank, N. J. A. Sloane, Modular and p-adic cyclic codes, *Des. Codes Cryptogr.*, **6**, 21–35 (1995).

[3] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, 1977.

[4] Josep Burillo, Class notes on lessons of algebraic structures, course 2012/13.

[5] Steven Roman, *Coding and Information Theory*, Springer-Verlag, 1992.

[6] Bernard R.McDonald, *Linear Algebra Over Commutative Rings*, Marcel Dekker, 1984.

[7] Josep M. Brunat Blay, Enric Ventura Capell, *Informació i codis*, Edicions UPC, 2001.

[8] Anton Betten, Michael Braun, et al., *Error-Correcting Linear Codes*, Springer, 2006.

[9] B. L. Van Der Waerden, *Modern algebra*, Frederick ungar publishing co. New York, 1953.

[10] M. F. Atiyah, I. G. Macdonald, *Introducción al algebra commutativa*, Editorial Reverte, 1980.