Chapter Based Lecture Notes

CO 331: Coding Theory

Prepared by: Calvin KENT
www.student.math.uwaterloo.ca/~c2kent/

Instructor: Alfred MENEZES
Term: Winter 2020

Last revised: 14 April 2020

Page

Table of Contents

i

Preface and Notation iii

0 Brief Overview of the Course 1

0.1 Big Picture . . . 1
0.2 Goal of Coding Theory and Classification of Codes . . . 1

1 Introduction and Fundamentals 3

Assumptions About the Communication Channel . . . 4
1.1 Decoding Strategy . . . 5
Nearest Neighbor Decoding . . . 6
Incomplete Maximum Likelihood Decoding (IMLD) . . . 6
Complete Maximum Likelihood Decoding (CMLD) . . . 6
Minimum Error Decoding (MED) . . . 6
1.2 Error Correcting and Decoding Capabilities of a Code . . . 7

2 Finite Fields 10

2.1 Introduction to Finite Fields . . . 10
2.2 Existence of Finite Fields . . . 12
2.3 Constructing Finite Fields . . . 15
Properties of Finite Fields . . . 16

3 Linear Codes 19

3.1 Properties of Linear Codes . . . 19
3.2 Dual Codes . . . 21
3.3 Properties of Linear and Dual Codes . . . 23
Decoding Linear Codes . . . 26
Decoding Algorithms for 1-error Correcting Codes . . . 26
General Decoding Problem for Binary Linear Codes . . . 27
Decoding Linear Codes in General . . . 27
Syndrome Decoding Algorithm . . . 28

4 The Binary Golay Code 30

The Binary Golay Code C_{23} (1949) . . . 30
The Extended Golay Code C_{24} . . . 30
Decoding Algorithm for C_{24} . . . 31
Reliability of C_{24} . . . 33

5 Cyclic Codes **35**

5.1 The Association Between $S \subseteq V_n(F)$ and $R = F[x]/(x^n - 1)$ . . . 35
Ideals of $R = F[x]/(x^n - 1)$ . . . 35
5.2 Constructing Cyclic Codes . . . 38
5.3 Dual Code of a Cyclic Code . . . 40
5.4 Syndromes in Cyclic Codes . . . 42
Burst Error Correcting . . . 44
Error Trapping Decoding for Cyclic Burst Errors . . . 45
Interleaving Codewords . . . 46
5.5 BCH Codes and Minimal Polynomials . . . 47
Minimal Polynomials . . . 47
Properties of Minimal Polynomials . . . 48
Formula for Calculating Minimal Polynomials . . . 49
⊛: Factoring $x^n - 1$ over GF(q) . . . 51
⊛: BCH codes . . . 51
⊛: BCH decoding . . . 51
⊛: Reed-Solomon codes . . . 51
⊛: Wrap-up . . . 51
⊛: Code-based Public-key Encryption (optional) . . . 51

Index **51**

Preface and Notation

This PDF document includes lecture notes for CO 331 - Coding Theory taught by Alfred MENEZES in Winter 2020.

For any questions contact me at `c2kent(at)uwaterloo(dot)ca`.

Notation

Throughout the course and the notes, unless otherwise is explicitly stated, we adopt the following conventions and notations.

- The university logo is used as a place holder.
- The textbook used in this class is `An Introduction to Error Correcting Codes with Applications`, by S.A. Vanstone and P.C. van Oorschot.
- Due to COVID-19, University of Waterloo took the following measures:
  - All activity for on-campus courses (e.g., classes, labs, tutorials, etc.) is suspended for one week. Course activity will stop on March 14. We will resume course activities in alternative formats on Monday, March 23.
  - At the end of the suspension of activities, the University is cancelling all in-person course activity, including in-person exams, through the end of the Winter Term on April 25.

  Detailed information regarding how the university handles COVID-19 can be found here: https://uwaterloo.ca/coronavirus/ and https://uwaterloo.ca/coronavirus/academic-information.

Calvin KENT

Chapter 0 – Brief Overview of the Course

In its broadest sense, coding theory deals with the reliable, efficient and secure transmission of data over channels that are subject to inadvertent noise and malicious intrusion.

0.1 Big Picture

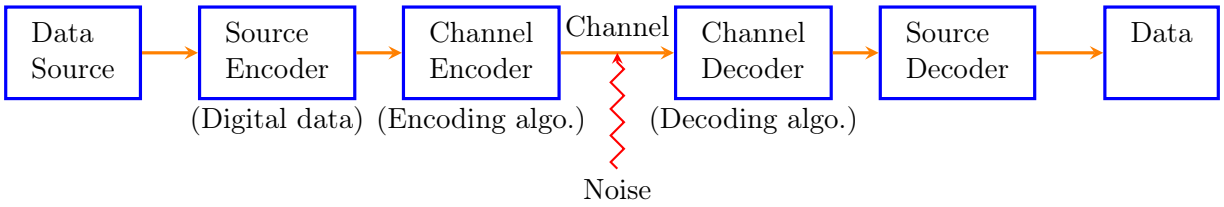

Figure 0.1.1: Big picture.

Example 0.1.1: Replication code example. Below TCD means that can be detected and TCC means that can be corrected.

Source Messages $\longrightarrow$ Codewords	# of errors TCD*	# of codewords TCC*	Rate
$0 \longrightarrow 0$ $1 \longrightarrow 1$	0	0	1
$0 \longrightarrow 00$ $1 \longrightarrow 11$	1	0	$\frac{1}{2}$
$0 \longrightarrow 000$ $1 \longrightarrow 111$	1	1	$\frac{1}{3}$
$0 \longrightarrow 00000$ $1 \longrightarrow 11111$	4	2	$\frac{1}{5}$
$0 \longrightarrow 0\ldots0$ $1 \longrightarrow \underbrace{1\ldots1}_{m \text{ times}}$	$m-1$	$\left\lfloor \frac{m}{2} \right\rfloor$	$\frac{1}{m}$

◁

0.2 Goal of Coding Theory and Classification of Codes

We want to design code so that we have

- high information rate,

- high error-correcting capability,
- with efficient encoding and decoding algorithms.

We group codes as follows.

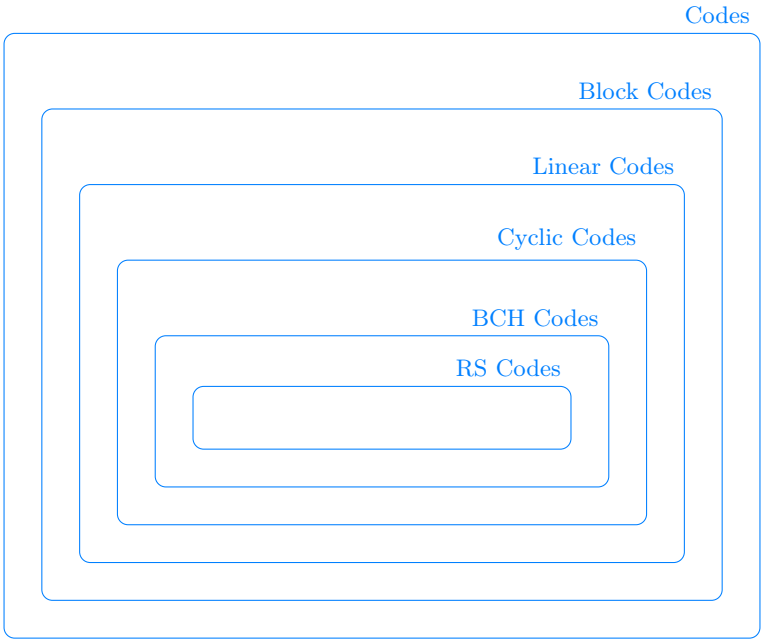

Figure 0.2.1: Classification of codes.

Things that will not be covered in this course:

- Flamming codes,
- Raptor codes,
- LDPC codes,
- Turbo codes.

Chapter 1 – Introduction and Fundamentals

Remark 1.0.1: We recall the big picture in Figure 0.1.1. In this course our focus is on what's going on between channel encoder and channel decoder, which is indicted by the purple circle.

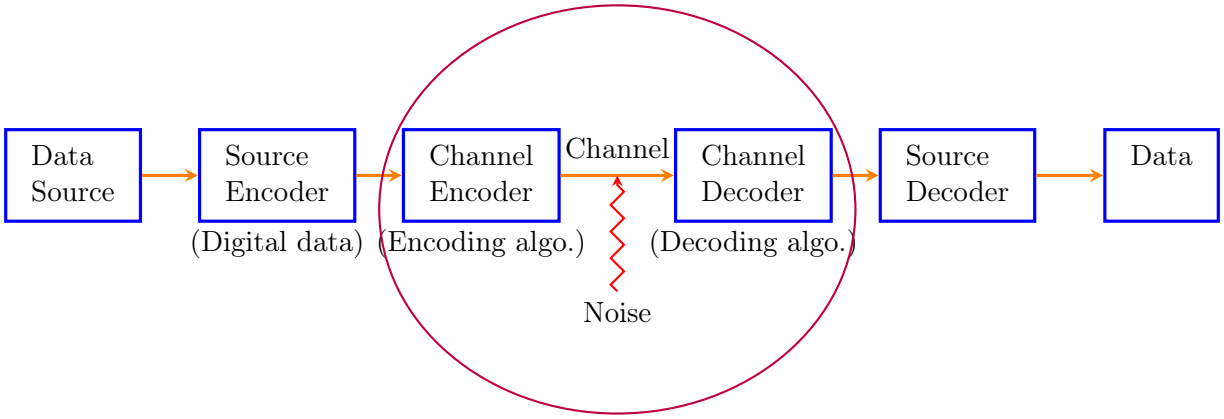

Figure 1.0.1: Course focus.

◁

Definition 1.0.2: We make the following definitions.

1. An ***alphabet*** A is a finite set of $|A| = q \geq 2$ symbols. e.g. $A = \{0, 1\}$.
2. A ***word*** is a finite sequence (*tuples* or *vectors*) of symbols from an alphabet A.
3. The ***length*** of a *word* is the number of symbols in it.
4. A ***code*** C over A is a finite set of words over A that has at least of length 2.
5. A ***codeword*** is a word in *code* C.
6. A ***block code*** is a code where all codewords have the same length. A block code C of length n containing M codewords over A is a subset $C \subseteq A^n$ with $|C| = M$. This is denoted by $[n, M]$ over A. ◁

Example 1.0.3: Let $A = \{0, 1\}$ be the binary alphabet and let $C = \{00000, 11100, 00111, 10101\}$. C is a $[5, 4]$ code over $\{0, 1\}$.

Messages	Codewords
00	00000
01	11100
10	00111
11	10101

Encoding is a one-to-one map. The channel encoder transmits only codewords but what's received by the channel decoder might not be a codeword.

As an example, suppose the channel decoder receives $r = 11001$. What should it do? In our

example, r is closest to 11100 and 10101. So it is more likely that the correct codeword was one of these two. However this may not be the case in practice. ◁

1.0.1 Assumptions About the Communication Channel

We have the following assumptions.

① The channel only transmits symbols from A.

② No symbols are deleted or added or transposed.

③ Errors are random.

Example 1.0.4: For $q = 2$ (***binary symmetric channel, BSC***) and $q = 3$ we have the encoding maps as follows.

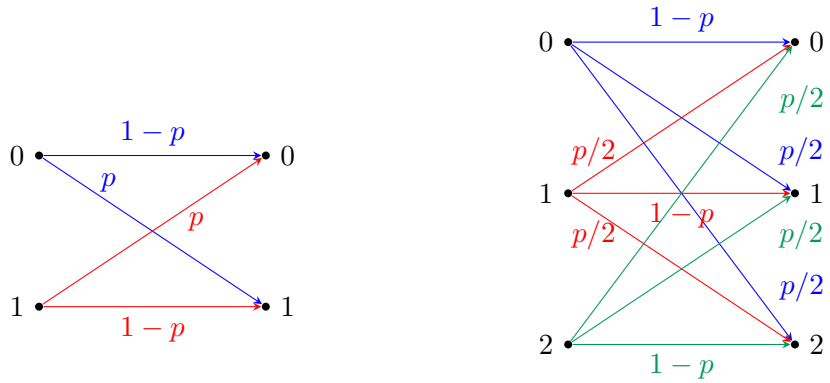

Figure 1.0.2: Encoding maps for $q = 2$ and $q = 3$ with probabilities.

Here we have p as the ***symbol error probability***. Hence, the probability of receiving the correct symbol is $1 - p$. Suppose the symbols transmitted are $X_1, X_2, \ldots$ and the symbols received are $Y_1, Y_2, \ldots$. Then, for all $i \geq 1$ and for all indices $1 \leq j, k \leq q$, we have the probability as

$$P(Y_i = a_j \mid X_i = a_k) = \begin{cases} 1 - p, \text{ if } j = k, \\ \dfrac{p}{q-1}, \text{ if } j \neq k. \end{cases}$$ ◁

Remark 1.0.5: We make the following remarks about BSC.

① If $p = 0$, the channel is perfect :)

② If $p = \frac{1}{2}$, the channel is useless :(

③ If $p > \frac{1}{2}$, then simply flip all bits that are received.

④ WLOG, we can assume $0 < p < \frac{1}{2}$.

⑤ Analogously, for a q-ary channel, we can assume that $0 < p < \frac{q-1}{q}$. ◁

Exercise 1.0.6: Verify remark ⑤ above. ◁

Definition 1.0.7: Let $x, y \in A^n$. The ***Hamming distance*** (simply referred as distance) $d(x, y)$ is the number of coordinate positions in which x and y differ. e.g. $d(10111, 01010) = 4$.

The distance of a code C is $d(C) = \min\{d(x, y) \mid x, y \in C \text{ and } x \neq y\}$. ◁

Theorem 1.0.8: d is a metric. That is, for all $x, y, z \in A^n$, Hamming distance satisfies the following.

1. (Positive definitiveness) $d(x, y) \geq 0$ and $d(x, y) = 0 \iff x = y$.
2. (Symmetry) $d(x, y) = d(y, x)$.
3. (Triangle Inequality) $d(x, z) \leq d(x, y) + d(y, z)$.

Proof: By definition distance cannot be negative, so we have $d(x, y) \geq 0$ for any $x, y \in A^n$. If $d(x, y) = 0$, then every i^{th} coordinate of x is same as every i^{th} coordinate of y. So, $x = y$ and if $x \neq y$, then x differs from y in at least one coordinate, so $d(x, y) \neq 0$. Hence, $d(x, y) = 0$ if and only if $x = y$, so distance is positive definite. If $d(x, y) = d$, then y and x differ in d coordinates, so $d(y, x) = d$. Now suppose, for contradiction, there exists $x, y, z \in A^n$ such that

$$d(x, z) > d(x, y) + d(y, z).$$

Let $d(x, z) = d$ and WLOG, suppose the coordinates in which x and z differ are the first d coordinates. We consider the possible values that $d(x, y)$ can take. Suppose $d(x, y) = 0$. Then $x = y$ but then $d(x, z) > d(y, z)$ is a contradiction. Suppose $d(x, y) = 1$. Then x and y differ in exactly 1 coordinate. If this coordinate is in the first d coordinates, then $d(y, z) \geq d - 1$. So,

$$d(x, z) = d > 1 + d(y, z) \geq d$$

is a contradiction. If the coordinates in which x and y differs is not in the first d coordinates, then $d(y, z) \geq d$ which also is a contradiction. Suppose $d(x, y) = k \geq 2$. If $k \geq d$ then we have a contradiction since distance of positive definite. If $k < d$, then by above argument $d(y, z) \geq d - k$ but then

$$d(x, z) = d > k + d(y, z) \geq k + d - k = d$$

is a contradiction. Hence, the triangle inequality is satisfied. □

Definition 1.0.9: The ***rate*** of an $[n, M]$-code C over A where $|A| = q$ is $R = \frac{\log_q M}{n}$. If the source messages all k-tuples over A, then $M = q^k$. So, we have

$$R = \frac{\log_q q^k}{n} = \frac{k}{n}.$$ ◁

Example 1.0.10: For $C = \{00000, 11100, 00111, 10101\}$ we have $R = \frac{2}{5}$ and $d(C) = 2$. ◁

1.1 Decoding Strategy

Let C be an $[n, M]$-code over A of distance d. Suppose some codeword is transmitted and $r \in A^n$ is received. Channel decoder has to decide the following.

1. No errors have occurred, accept r.
2. Errors have occurred and (decode) correct r to some codeword.
3. Erros have occurred and correction process is not possible.

1.1.1 Nearest Neighbor Decoding

1.1.1.1 Incomplete Maximum Likelihood Decoding (IMLD)

IMLD corrects r to be the unique codeword $c_r \in C$ for which the distance between r and c_r, $d(r, c_r)$, is smallest. i.e. $d(r, c_r) < d(r, c)$ for all $c \in C$ with $c \neq c_r$. If C is not unique codeword, r is rejected.

1.1.1.2 Complete Maximum Likelihood Decoding (CMLD)

Same as IMLD, except ties are broken arbitrarily.

We want to know if IMLD is a reasonable strategy.

Theorem 1.1.1: IMLD selects the codeword C that maximizes the conditional probability $P(r|C)$, the probability that r is received given that C was sent.

Before starting the proof we recall Bayes theorem from statistics.

Recall 1.1.2: Bayes theorem states that for two events A and B,

$$P(B)P(A|B) = P(A)P(B|A). \qquad \triangleleft$$

Proof: Suppose $c_1, c_2 \in C$ with $d(c_1, r) = d_1$ and $d(c_2, r) = d_2$. Suppose $d_1 > d_2$. So c_1 is further away from r than c_2. Then IMLD selects c_2. So we are only required to show $P(r|c_2) > P(r|c_1)$. We have

$$P(r|c_1) = (1-p)^{n-d_1}\left(\frac{p}{q-1}\right)^{d_1},$$
$$P(r|c_2) = (1-p)^{n-d_2}\left(\frac{p}{q-1}\right)^{d_2}.$$

Hence,

$$\frac{P(r|c_1)}{P(r|c_2)} = (1-p)^{d_2-d_1}\left(\frac{p}{q-1}\right)^{d_1-d_2} = \left(\frac{p}{(1-p)(q-1)}\right)^{d_1-d_2}.$$

We also have

$$p < \frac{q-1}{q} \implies pq < q-1 \implies 0 < q - pq - 1 \implies p < p + q - pq - 1 = (1-p)(q-1)$$
$$\implies \frac{p}{(1-p)(q-1)} < 1.$$

Since $d_1 > d_2$, then, $\frac{P(r|c_1)}{P(r|c_2)} < 1$. Hence, $P(r|c_1) < P(r|c_2)$. $\square$

1.1.1.3 Minimum Error Decoding (MED)

The ideal strategy is to correct r to $c \in C$ that maximizes $P(c|r)$. This is called Minimum Error Decoding (MED).

Remark 1.1.3: IMLD is not the same as MED. Let $C = \{000, 111\}$ where $c_1 = 000$ and $c_2 = 111$ and where the alphabet is $A = \{0, 1\}$. So we have

$$0 \to 000, \text{ and } 1 \to 111.$$

Suppose $P(c_1) = 0.1$ and $P(c_2) = 0.9$. Suppose $p = 0.25$ and $r = 100$.

- With IMLD we get $r \to 000$ since 100 is closer to 000 than it is to 111.
- We now check to see what MED gives us. By Bayes theorem we have

$$P(c_1|r) = \frac{P(r|c_1)P(c_1)}{P(r)} = \frac{p(1-p)^2 0.1}{P(r)} = \frac{9}{640P(r)},$$
$$P(c_2|r) = \frac{P(r|c_2)P(c_2)}{P(r)} = \frac{p^2(1-p)0.9}{P(r)} = \frac{27}{640P(r)}.$$

  So, $P(c_2|r) > P(c_1|r)$. Hence, MED picks c_2 and gives us $r \to 111$. ◁

Remark 1.1.4: We make the following remarks.

① IMLD selects c such that $P(r|c)$ is maximum.

② MED selects c such that $P(c|r)$ is maximum.

③ MED has a drawback that required knowledge of the $P(c_i)$ for $i = 1, \ldots, M$.

④ Suppose source messages are equally likely. So $P(c_i) = \frac{1}{M}$ for each $i = 1, \ldots, M$. Then,

$$P(r|c_i) = \frac{P(c_i|r)P(r)}{P(c_i)} = P(c_i|r) \cdot \underbrace{\frac{P(r)}{M}}_{\text{does not depend on } i}.$$

  So, maximizing $P(r|c_i)$ is same as maximizing $P(c_i|r)$. Hence, IMLD is same as MED.

In the remainder of the course we will use IMLD/CMLD. ◁

1.2 Error Correcting and Decoding Capabilities of a Code

- If C is used for error correction, the strategy is IMLD/CMLD.
- If C is used for error detection only, the strategy is to reject r if $r \notin C$, otherwise accept it.

Definition 1.2.1: A code C is called an ***e-error correcting code*** if the decoder always makes the correct decision if at most e errors per codeword are introduced. We define ***e-error detecting code*** similarly. ◁

Example 1.2.2:

① $C = \{0000, 1111\}$ is a 1-error correcting code but not a 2-error correcting code.

② $C = \{\underbrace{00\ldots0}_{m \text{ times}}, \underbrace{11\ldots1}_{m \text{ times}}\}$ is a $\left\lfloor \frac{m-1}{2} \right\rfloor$-error correcting code.

③ $C = \{0000, 1111\}$ is a 3-error detecting code. ◁

Theorem 1.2.3: Let $d(C) = d$. Then C is a $(d-1)$-error detecting code.

Proof: Suppose $c \in C$ is transmitted and r is received. If no errors occur (that is, $d(r,c) = 0$) then $r = c \in C$ and decoder accepts r. If d or more errors occur, then the decoder can make the wrong choice since $d(C) = d$. If at least 1 but less than d errors occur, then $1 \leq d(r,c) \leq d-1$. In this case $r \notin C$ and the decoder rejects r. Hence, C is a $(d-1)$-error detecting code. □

Corollary 1.2.4: If $d(C) = d$, then C is not a d-error detecting code.

Proof: Since $d(C) = d$, then $\exists\ c_1, c_2 \in C$ such that $d(c_1, c_2) = d$. If c_1 is sent and r is received and with d errors, then it is possible that $r = c_2$. In this case the decoder accepts c_2 but this is the wrong procedure. Hence C is not a d-error detecting code. □

Theorem 1.2.5: If $d(C) = d$, then C is a $\lfloor \frac{d-1}{2} \rfloor$-error correcting code.

Proof: Suppose $c \in C$ is transmitted and at most $\lfloor \frac{d-1}{2} \rfloor$ errors are introduced and r is received. Let $c_1 \in C$ be arbitrary with $c_1 \neq c$. By triangle inequality we have

$$d(c, c_1) \leq d(c, r) + d(r, c_1) \implies d(r, c_1) \geq d(c, c_1) - d(c, r) \geq d - \frac{d-1}{2} = \frac{d+1}{2} > \frac{d-1}{2}.$$

So, c is the unique codeword closest to r. So, IMLD,CMLD will decode r to c and hence C is a $\lfloor \frac{d-1}{2} \rfloor$-error correcting code. □

Theorem 1.2.6: If $d(C)$ then C is not a $\lfloor \frac{d-1}{2} \rfloor + 1$-error correcting code.

Proof: Exercise. ◁

Remark 1.2.7: Given an alphabet A with size q, n, M and d for a block code, does there exists an $[n, M]$-code C over A with $d(C) = d$?

Consider the set of all possible n-tuples over A and let $C = \{c_1, \ldots, c_m\}$. Let $e = \lfloor \frac{d-1}{2} \rfloor$. For any codeword $c \in C$, define

$$S_c = \{r \in A^n \mid d(r, c) \leq e\} = \text{Sphere of radius } e \text{ centered at } c.$$

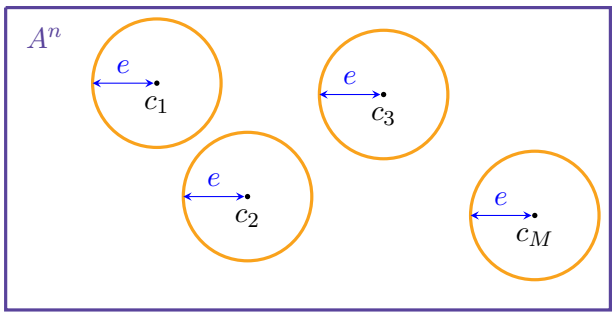

Figure 1.2.1: S_{c_i} are shown in orange for $i = 1, 2, 3, M$.

In theorems above we proved that given distinct codewords c_1, c_2, the spheres around them don't overlap. So $S_{c_i} \cap S_{c_j} = \varnothing$ where $c_i \neq c_j$. Thus the question can be viewed as a *sphere packing*

problem.

This question of asking if we can place M spheres of radius e in A^n such that no two spheres overlap is a purely combinatorial problem. ◁

Example 1.2.8: Given $A = \{0, 1\}$, $n = 128$, $M = 2^{64}$ codewords, determine if a $[n, M]$-code C over A with $d(C) = d$ exists.

The answer is yes and we will show how to get and use this answer in the following lectures. ◁

Remark 1.2.9: In the following weeks we will view set of all words, A^n, as a vector space of dimension n over the field $\mathbb{Z}_q$ where $|A| = q$. We will choose the code C to be a M-dimensional subspace of this vector space and we will choose special subspaces that satisfy the $d(C) = d$ requirement. ◁

Chapter 2 – Finite Fields

2.1 Introduction to Finite Fields

Remark 2.1.1: We denote the multiplication operation, $\times$ as $\cdot$ or concatenation. It'll be always clear from the context. ◁

Definition 2.1.2: A ***binary operation*** on a set S is a function $* : S^2 \to S$ where $S^2 = S \times S = \{(a, b) \mid a, b \in S\}$. We usually write $a * b$ instead of $*(a, b)$. ◁

Definition 2.1.3: A ***ring*** is a set R equipped with two binary operations addition $(+)$ and multiplication $(\cdot)$ and contains an element 0 such that

1. $+$ is associative, i.e. $(a + b) + c = a + (b + c)$ for all $a, b, c \in R$,
2. $+$ is commutative, i.e. $a + b = b + c$ for all $a, b, c \in R$,
3. 0 is additive identity, i.e. $a + 0 = 0 + a = a$, for all $a \in R$
4. every $a \in R$ has an additive inverse, i.e. there exists $-a \in R$ such that $a+(-a) = (-a)+a = 0$,
5. $\times$ is associative, i.e. $(ab)c = a(bc)$ for all $a, b, c \in R$,
6. $\times$ is distributive over $+$, i.e. $a(b + c) = ab + ac$ and $(a + b)c = ac + bc$ for all $a, b, c \in R$. ◁

Definition 2.1.4: Let R be a ring.

1. If $\times$ is commutative in R, that is, if $ab = ba$ for all $a, b \in R$, then we say R is ***commutative***.
2. If R contains additive identity, that is, if $R \ni 1 \neq 0$ such that $1a = a1 = a$ for all $a \in R$, then we say R ***has an identity*** (or that R has a 1). 1 is also referred as the multiplicative identity.
3. If R has a 1 and if there exists $a, b \in R$ such that $ab = 1$, we say a is a ***invertible*** (or that a is a ***unit***). Note that if there exists $a \in R$ such that $a \neq 1$ and a is a unit, then there exists $b \in R$ such that $b \neq 1$ and $ab = 1$. i.e. the multiplicative identity is unique.
4. If R has 1 and if for all non-zero $a \in R$ is a unit then R is called a ***division ring***.
5. If R is commutative and is a division ring then it is called a ***field***. Generally we use F or $\mathbb{F}$ to denote fields. ◁

Definition 2.1.5: We say a field F is infinite if $|F| \geq \aleph_0$. i.e. if $|F|$ is not finite. Otherwise we say F is finite. ◁

Example 2.1.6: Let $n \in \mathbb{Z}^+$. The sets $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ and $\mathbb{Z}_n$ are all commutative rings with 1 but only $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ and $\mathbb{Z}_p$ for p is prime are fields. ◁

Definition 2.1.7: The ***order of a field*** F is its cardinality. The ***multiplicative order of an element*** $a \in F$ is the smallest positive integer n that satisfies $a^n = 1$. We define the ***additive order*** similarly. ◁

Recall 2.1.8: Let $n \geq 2$ be a positive integer. The integers modulo n denoted by $\mathbb{Z}_n$ or $\mathbb{Z}/n\mathbb{Z}$ or $\mathbb{Z}/(n)$ is the set of all equivalence classes modulo n. In this course we will use $\mathbb{Z}_n$. That is,

$$\mathbb{Z}_n \underset{\text{def}}{=} \{[0], [1], \ldots, [n-1]\}.$$

Here the square brackets denote equivalence classes modulo n. That is, if $a \in [b]$, then $a = b \mod n$. The addition and multiplication is defined in the usual sense in between equivalence classes. i.e. $[a] + [b] = [a + b]$ and $[a][b] = [ab]$. For convenience we will drop the square brackets when talking about $\mathbb{Z}_n$. ◁

Example 2.1.9: The set of integers modulo 9 is $\mathbb{Z}_9 = \{0, 1, \ldots, 8\}$ and $7 + 8 = 6 \mod 9 = [6] = 6$ and $7 \cdot 8 = 2 \mod 9 = [2] = 2$. ◁

Theorem 2.1.10: Let $n \geq 2$ be a positive integer. $\mathbb{Z}_n$ is a field if and only if n is prime.

Proof: Suppose n is composite. So $n = ab$ where $a, b \in \mathbb{Z}^+$ with $2 \leq a, b \leq n - 1$. Suppose, for contradiction, a is a unit. Then $\exists\ a^{-1} \in \mathbb{Z}_n$ such that $aa^{-1} = 1 \mod n$. Thus, $aba^{-1} = na^{-1} = b \mod n$. Hence, $b = 0 \mod n$. Then $n|b$ which is a contradiction. Hence, if n is composite then $\mathbb{Z}_n$ is not a field. The contrapositive of this argument shows if $\mathbb{Z}_n$ is a field then n is prime. Now suppose n is prime. Since $\mathbb{Z}_n$ is a commutative ring with 1 we only need to show $\mathbb{Z}_n$ is a division ring. Then $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. Let $a \in \mathbb{Z}_n$ be non-zero. Then $1 \leq a \leq n - 1$. Since n is prime then $\gcd(a, n) = 1$. Then by Bezout's identity there exists $s, t \in \mathbb{Z}$ such that $as + nt = 1$. Hence, $as + nt = as \mod n$. Hence, $a^{-1} = s$. Since a was arbitrary, then every non-zero element in $\mathbb{Z}_n$ has an inverse. Hence $\mathbb{Z}_n$ is a field. □

Remark 2.1.11: By above theorem we see that there exist fields with order p where p is prime. Do there exist fields of orders 4 and 6?
The answer is yes and we will see more about such fields in the following lectures. ◁

Definition 2.1.12: The ***characteristic of a field*** F, denoted by $\text{char}(F)$, is the smallest positive integer m such that $\underbrace{1 + 1 + \cdots + 1}_{m \text{ times}} = 0$. If no such exists, then we say the characteristic is zero. ◁

Example 2.1.13: $\text{char}(\mathbb{Q}) = 0 = \text{char}(\mathbb{R}) = \text{char}(\mathbb{C})$ and $\text{char}(\mathbb{Z}_p) = p$. ◁

Theorem 2.1.14: Let F be a field. If $\text{char}(F) = 0$ then F is not finite.

Proof: We have $1, 1 + 1, \underbrace{1 + 1 + \cdots + 1}_{m \text{ times}} \in F$ for all $m \in \mathbb{Z}^+$. It is sufficient to show all $1, 1 + 1, 1 + 1 + \cdots + 1$ are distinct. Suppose, for contradiction, there exists distinct $m, n \in \mathbb{Z}^+$ such that

$$\underbrace{1 + \cdots + 1}_{m \text{ times}} = \underbrace{1 + \cdots + 1}_{n \text{ times}}.$$

WLOG, suppose $m > n$. Then,

$$\underbrace{1 + \cdots + 1}_{m \text{ times}} = \underbrace{1 + \cdots + 1}_{n \text{ times}} + \underbrace{1 + \cdots + 1}_{m-n \text{ times}} = \underbrace{1 + \cdots + 1}_{n \text{ times}}.$$

Hence, $\underbrace{1 + \cdots + 1}_{m-n \text{ times}} = 0$. But this means $\text{char}(F) = m - n \in \mathbb{Z}^+$ which is a contradiction. Hence F is infinite. □

Theorem 2.1.15: If F is a finite field then $\text{char}(F)$ is prime.

Proof: Suppose $\text{char}(F) = m$ where m is composite. Then, $m = ab$ where $2 \leq a, b \leq m-1$. Hence we have

$$\underbrace{1 + \cdots + 1}_{m \text{ times}} = \left(\underbrace{1 + \cdots + 1}_{a \text{ times}}\right)\left(\underbrace{1 + \cdots + 1}_{b \text{ times}}\right) = 0.$$

Note that we also have $\underbrace{1 + \cdots + 1}_{a \text{ times}}, \underbrace{1 + \cdots + 1}_{b \text{ times}} \in F$. Since $\text{char}(F) = m > a$ then there exists $c \in F$ such that

$$c \cdot \underbrace{1 + \cdots + 1}_{a \text{ times}} = 1.$$

This gives us

$$c \cdot \left(\underbrace{1 + \cdots + 1}_{a \text{ times}}\right)\left(\underbrace{1 + \cdots + 1}_{b \text{ times}}\right) = c \cdot \underbrace{1 + \cdots + 1}_{m \text{ times}} = 0 = 1 \cdot \underbrace{1 + \cdots + 1}_{b \text{ times}}.$$

But then $b = 0$ which is a contradiction. Hence $\text{char}(F)$ is prime. □

Remark 2.1.16: In the following weeks we will consider a finite field F with order n and characteristic p for some prime p. We will see that $\mathbb{Z}_p$ is a subfield of F and F is a vector space over $\mathbb{Z}_p$, say of dimension k. We will see that $n = p^k$. ◁

2.2 Existence of Finite Fields

Definition 2.2.1: We say two fields F_1 and F_2 are ***isomorphic*** if they have the same binary operations and if there exists a bijection $\alpha : F_1 \to F_2$ that preserves the operations. i.e.

$$\begin{aligned} \alpha(a+b) &= \alpha(a) + \alpha(b) \in F_2, \\ \alpha(ab) \quad &= \alpha(a)\alpha(b) \quad \in F_2, \quad \forall\, a, b \in F_1. \end{aligned}$$

◁

Definition 2.2.2: Let F be a field. A subset $S \subseteq F$ is called a ***subfield*** of F if S is also a field using the same operations used in F. ◁

Example 2.2.3: Let F be a finite field with characteristic p. Consider $F \supseteq E = \{0, 1, 1+1, \underbrace{1+1+\cdots+1}_{p-1 \text{ times}}\}$.
We see that the set E equipped with the field operations of F is also a field. The order of E is p. If we label the elements of E in a natural way such that $\underbrace{1+1+\cdots+1}_{n \text{ times}} = n$, we have

$$F \supseteq E = \{\underbrace{0}_{0 \text{ times}}, \underbrace{1}_{1 \text{ times}}, \underbrace{1+1}_{2 \text{ times}}, \ldots, \underbrace{1+1+\cdots+1}_{p-1 \text{ times}}\} \cong \{0, 1, 2, \ldots, p-1\} = \mathbb{Z}_p.$$

So we can say E is isomorphic to $\mathbb{Z}_p$. ◁

Theorem 2.2.4: If F is a finite field of characteristic p, then $\mathbb{Z}_p$ is a subfield of F.

Proof: Exercise. ◁

Remark 2.2.5: Consider $\mathbb{Z}_p \subseteq F$ for a finite field F. We make the following remarks.

1. Each $f \in F$ is a vector.
2. Each $k \in \mathbb{Z}_p$ is a scalar.
3. Addition in F is defined by vector addition.
4. Multiplication in F by elements in $\mathbb{Z}_p$ is defined by scalar multiplication. ◁

Theorem 2.2.6: If F is a finite field of characteristic p, then F is a vector space over $\mathbb{Z}_p$.

Proof: Exercise. ◁

Theorem 2.2.7: If F is a finite field of characteristic p, then $|F| = n = p^k$ for some $k \in \mathbb{N}$.

Proof: Since F is a finite field then by Theorem 2.2.6 F is a vector field over $\mathbb{Z}_p$. Then, $\dim F = k$ for some $k \in \mathbb{Z}^+$. Let $\{\alpha_1, \ldots, \alpha_k\}$ be a basis of F. Then, every element in f can be uniquely expressed as a linear combination of the elements in this basis. That is, for all $a \in F$ we have

$$a = c_1\alpha_1 + \cdots + c_n\alpha_n \quad \text{where} \quad c_i \in \mathbb{Z}_p \text{ for } i = 1, \ldots, n.$$

Note that the linear combinations of $c_i\alpha_i$ uniquely determine a. Since F is also a field, then it's closed under addition and multiplication. Hence, every linear combination of $c_i\alpha_i$ is also an element of F. For each α_i, there are p possible choices for c_i. Since the basis contains k vectors, then F has p^k elements. Hence, $|F| = p^k$. □

Example 2.2.8: There are no fields of order 6. ◁

Remark 2.2.9: How can we show if there a fields of orders $4, 8, 9$?

We consider $\mathbb{Z}$, which is an infinite commutative ring. We pick $n \geq 2$ and consider the congruence relation on $\mathbb{Z}$, that is $a = b(\mod n)$. The set of equivalence classes for the congruence equivalence relation forms a finite set $\mathbb{Z}_n$ and for n prime, $\mathbb{Z}_n$ forms a finite field of order n.

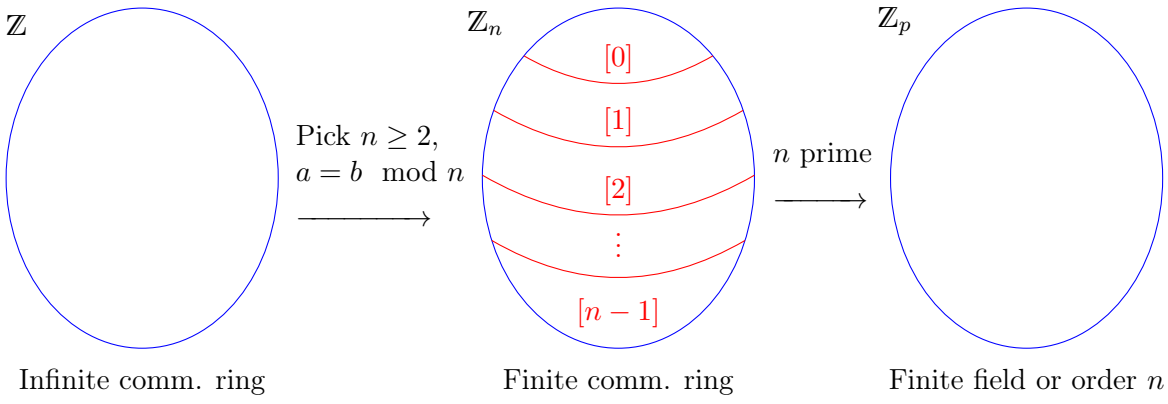

Figure 2.2.1: Constructing a finite field from infinite commutative ring $\mathbb{Z}$.

Here $[0], [1], \ldots, [n-1]$ are sets of equivalence classes for modulo n. We can also consider $\mathbb{Z}_p[x]$ (polynomial ring over $\mathbb{Z}_p$) and do the same procedure as before, except by picking the congruence relation over polynomials for some polynomial f such that $\deg f \geq 1$ and by picking the irreducible polynomials in the last step.

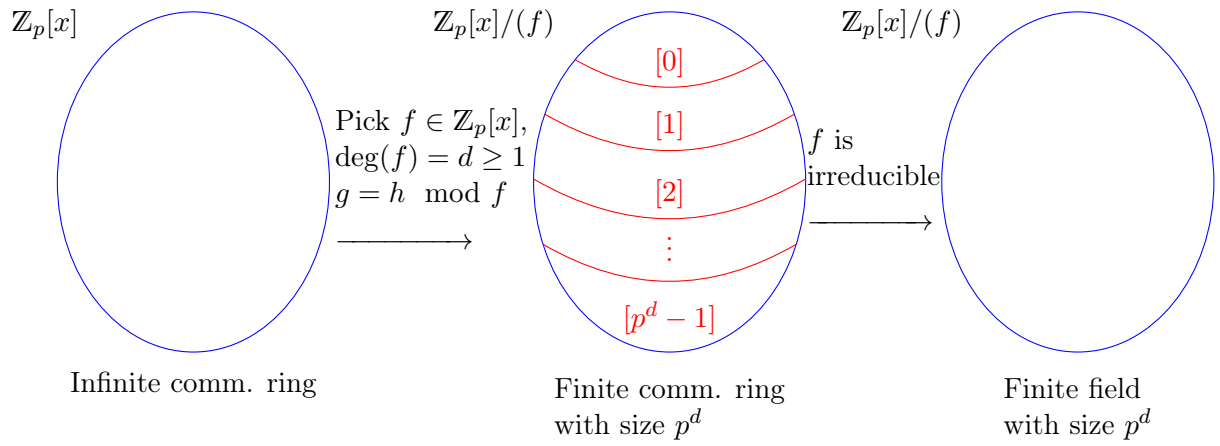

Figure 2.2.2: Constructing finite field from infinite commutative ring $\mathbb{Z}_p[x]$.

Here $[0], [1], \ldots, [p^d - 1]$ are sets of equivalence classes in polynomials for $f \in \mathbb{Z}_p[x]$. ◁

Definition 2.2.10: Let F be a field. The ***set of all polynomials in*** x ***over*** F (polynomials with coefficients from F) is denoted as $F[x]$. Addition and multiplication operations are defined in the usual way with coefficient arithmetic in F. Note that generally speaking, $F[x]$ is not always a field since $x \in F[x]$ can have no inverse but $F[x]$ is always a commutative ring. ◁

Example 2.2.11: In $\mathbb{Z}_{11}[x]$, we have $f = 2 + 5x + 6x^2 \in \mathbb{Z}_{11}[x]$ and $g = 3 + 9x + 5x^2 \in \mathbb{Z}_{11}[x]$ and $f + g = 5 + 3x \in \mathbb{Z}_{11}[x]$. ◁

Theorem 2.2.12: Let F be a field. Then, $F[x]$ is a commutative ring.

Proof: Since F is a field, then it's a commutative division ring. So, every $a \in F$ has an inverse. Since in $F[x]$ addition and multiplication is defined the same way as they do in F, then $F[x]$ is a commutative ring. Since $x \in F[x]$ but x is not a unit in $F[x]$, then $F[x]$ is not a field. □

Definition 2.2.13: Let F be a field and let $f \in F[x]$ with $\deg f \geq 1$. If $g, h \in F[x]$ and if $f | g - h$, then we write $g = h \mod f$. Note that this is equivalent to saying if $g = h \mod f$ then $g - h = \ell f$ for some $\ell \in F[x]$.

For zero polynomial, we say $\deg 0 = -\infty$ to be consistent with the division algorithm. ◁

Theorem 2.2.14: We state the following theorems regarding what we covered in this section.

1. Congruency is an equivalence relation.
2. Let $g \in F[x]$. The equivalence class containing g, $[g]$, is well-defined. That is,
$$[g] = \{h \in F[x] \mid h = g \mod f\}.$$
3. Addition and multiplication of equivalence classes for congruency in polynomials is well-defined.
4. The set of all equivalence classes, denoted by $F[x]/(f)$ where $f \in F[x]$ with $\deg f \geq 1$ is a commutative ring.

⑤ The polynomials in $F[x]$ of degree less than $\deg f$ are a system of distinct representatives of the equivalence classes in $F[x]/(f)$.

Proof: We only show ⑤ and leave others as exercise. Let $g \in F[x]$. By division algorithm in polynomials we can write $g = \ell f + r$ where $\deg r < \deg f$. So, $g - r = \ell f$. Hence, $g = r \mod f$. Hence, $[g] = [r]$ and we have $\deg r < \deg f$. Also, if $r_1, r_2 \in F[x]$ where $r_1 \neq r_2$ with $\deg r_i < \deg f$ for $i = 1, 2$ then, $f \not| r_1 - r_2$. So, $r_1 \neq r_2 \mod f$. Hence, $[r_1] \neq [r_2]$. □

2.3 Constructing Finite Fields

Last class we proved a system of distinct representatives for $\mathbb{Z}_p/(f)$ is $[r(x)]$ where $r \in \mathbb{Z}_p[x]$ and $\deg r < \deg f = n$. This gave us $|\mathbb{Z}_p[x]/(f)| = p^n$.

Definition 2.3.1: Let F be a field and $f(x) \in F[x]$ with $\deg f = n \geq 1$. Then, we say f is ***irreducible*** over F if f cannot be written as $f = gh$ where $g, h \in F[x]$ and $\deg g, \deg h \geq 1$. ◁

Example 2.3.2: $f(x) = x^2 + 1$ is irreducible over $\mathbb{R}, \mathbb{Z}_3$ but it's reducible over $\mathbb{C}$ and $\mathbb{Z}_2$ since $x^2 + 1 = (x - i)(x + i)$ and $1^2 + 1 = 0 \mod 2$. ◁

Theorem 2.3.3: Let F be a field and let $f \in F[x]$ of degree $n \geq 1$. Then, the ring of polynomials over F modulo f, $F[x]/(f)$, is a field if and only if f is irreducible over F.

Proof: We first note that $F[x]/(f)$ is a commutative ring for any f with $\deg f \geq 1$ since $F[x]$ is a field and addition and multiplication operations in F are also commutative in polynomials. For the backward direction suppose $g \in F[x]/(f)$ with $g \neq 0$ and $\deg g < \deg f$. Then, $\gcd(g, f) = 1$. So, by the extended Euclidean algorithm for polynomials, there exists $s, t \in F[x]$ such that $gs + ft = 1$. Then,

$$1 = gs + ft = gs \mod f.$$

So, $s = g^{-1}$. Since this is true for any non-zero $g \in F[x]/(f)$, then $F[x]/(f)$ is a commutative division ring and hence by definition, it's a field. □

Exercise 2.3.4: Prove forward direction. ◁

Remark 2.3.5: To construct a finite field of order p^n where $n \geq 2$, we need an irreducible polynomial $f \in \mathbb{Z}_p[x]$ of degree n. This gives us a finite field $\mathbb{Z}_p[x]/(f)$ with order p^n. ◁

Theorem 2.3.6: For any prime $p \in \mathbb{Z}$ and integer $n \geq 2$, there exists an irreducible polynomial of degree n over $\mathbb{Z}_p$.

Proof: The proof can be done using generating functions from MATH 249 and it's beyond the scope of this course. □

Theorem 2.3.7: There exists a finite field of order q if and only if q is a prime power.

Proof: Exercise. □

Example 2.3.8: To construct a finite field or order $4 = 2^2$, we can take $f(x) = x^2 + x + 1$ which is irreducible over $\mathbb{Z}_2[x]$. We have $\mathbb{Z}_2[x]/(x^2 + x + 1) = \{0, 1, x, x + 1\}$ where $1^{-1} = 1, x^{-1} = x + 1$.

To construct a finite field or order $8 = 2^3$, we can take $f(x) = x^3 + x + 1$ or $g(x) = x^3 + x^2 + 1$ since

both are irreducible over $\mathbb{Z}_2[x]$. We have

$$\mathbb{Z}_2[x]/(f) = \{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}, \quad \text{where} \quad \begin{aligned} x^{-1} &= x^2+1, \\ (x+1)^{-1} &= x^2+x, \\ (x^2)^{-1} &= x^2+x+1 \end{aligned}$$

Note that we also have $\mathbb{Z}_2[x]/(g) = \{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$ but $\mathbb{Z}_2[x]/f \neq \mathbb{Z}_2[x]/g$. It is easy to see this since the zero elements in these fields represent different things but $\mathbb{Z}_2[x]/f \cong \mathbb{Z}_2[x]/g$ since there exists a bijection between these two fields. ◁

Recall 2.3.9: Let F_1, F_2 be fields. We say F_1 is isomorphic to F_2 if there exists a bijection $\alpha : F_1 \to F_2$ that preserves the operations. i.e.

$$\begin{aligned} \alpha(a+b) &= \alpha(a) + \alpha(b) \in F_2, \\ \alpha(ab) \quad &= \alpha(a)\alpha(b) \quad \in F_2, \quad \forall\, a, b \in F_1. \end{aligned} \quad \triangleleft$$

Theorem 2.3.10: Any two finite fields with same order are isomorphic.

Proof: Exercise. □

Notation 2.3.11: We denote finite fields with order q as $\mathrm{GF}(q)$ (Galois field). In above examples, we show two different representations of $\mathrm{GF}(2^3)$. ◁

Recall that by Remark 2.3.5 a finite field of order q, $\mathrm{GF}(q)$, exists if and only if $q = p^n$ for some prime $p \in \mathbb{Z}$ and $n \geq 1$. If $\mathrm{GF}(q) = \mathbb{Z}_p[x]/(f)$, then $\operatorname{char} \mathbb{Z}_p[x]/(f) = p$.

Example 2.3.12: To construct $\mathrm{GF}(16) = \mathrm{GF}(2^4)$, we need an irreducible polynomial of degree 4 in $\mathbb{Z}_2$. We can take $f(x) = x^4 + x + 1$. Note that $x^2 + 1$ is the only irreducible polynomial with degree 2 in $\mathbb{Z}_2$. So, f is also irreducible over $\mathbb{Z}_2$ since $f(0), f(1) \neq 0 \mod 2$ and $x^2 + 1$ is not a factor of f by long division. So we have $\mathrm{GF}(16) = \mathbb{Z}_2[x]/(f)$. ◁

2.3.1 Properties of Finite Fields

Before we state and prove Frosh's Dream theorem, we prove the following lemma.

Lemma 2.3.13: If F and K are two finite fields that are isomorphic, then $\operatorname{char} F = \operatorname{char} K$.

Proof: Let $\operatorname{char} F = p$. Then, $\underbrace{1_F + \cdots + 1_F}_{p \text{ times}} = 0$. Let $\alpha : F \to K$ be a bijection. For any $a \in F$ we have

$$\alpha(a \cdot 1_F) = \alpha(a)\alpha(1_F) = \alpha(a), \text{ where } 1_F \text{ is the identity in } F.$$

So, $\alpha(1_F) = 1_K$ is the identity in K. We also have

$$\alpha(1_F + \cdots + 1_F) = \underbrace{\alpha(1_F) + \cdots + \alpha(1_F)}_{p \text{ times}} = 0.$$

Hence $\operatorname{char}(K) | p$. Hence $\operatorname{char} K = p$. □

Theorem 2.3.14 (Frosh's Dream)**:** Let $\alpha, \beta \in \mathrm{GF}(q)$ with $|\mathrm{GF}(q)| = p$. Then, $(\alpha+\beta)^p = \alpha^p + \beta^p$.

Proof: By binomial theorem we have

$$(\alpha+\beta)^p = \sum_{i=0}^{p} \binom{p}{i} \alpha^i \beta^{p-i} = \alpha^p + \sum_{i=1}^{p-1} \binom{p}{i} \alpha^i \beta^{p-i} + \beta^p.$$

Note that for prime $p \in \mathbb{Z}$, we have $p \not| \, 1 \cdot 2 \cdots p-1$. So, for all $0 \leq i < p$, we have $p \not| \, i!$. So, we have

$$\binom{p}{i} = \frac{p \cdot (p-1)!}{i!(p-i)!} = p \cdot \gamma, \quad \text{where} \quad \gamma \in \mathbb{N} \text{ with } \gamma \neq 0 \text{ and } p \not| \, \gamma.$$

This gives us

$$\sum_{i=1}^{p-1} \binom{p}{i} \alpha^i \beta^{p-i} = \sum_{i=1}^{p-1} p\gamma_i \alpha^i \beta^{p-i}.$$

Since $|\mathrm{GF}(q)| = p$, then $\mathrm{GF}(q) \cong \mathbb{Z}_p$. Hence, $\operatorname{char} \mathrm{GF}(q) = p$. This gives us

$$\sum_{i=1}^{p-1} \binom{p}{i} \alpha^i \beta^{p-i} = \sum_{i=1}^{p-1} p\gamma_i \alpha^i \beta^{p-i} = \sum_{i=1}^{p-1} \underbrace{(1+\cdots+1)}_{p \text{ times}} \gamma_i \alpha^i \beta^{p-i} = 0.$$

Hence, $(\alpha+\beta)^p = \alpha^p + \beta^p$. □

Remark 2.3.15: Frosh's Dream theorem is also known as Anti-Calculus Lemma and more generally for $\alpha, \beta \in \mathrm{GF}(q)$ we have $((\alpha+\beta)^p)^m = (\alpha^p)^m + (\beta^p)^m$. ◁

Theorem 2.3.16: Let $\alpha \in \mathrm{GF}(q)$. Then $\alpha^q = \alpha$.

Proof: If $\alpha = 0$, then $\alpha^q = 0 = \alpha$. If $\alpha \neq 0$, then let $\mathrm{GF}(q)^* = \{\alpha_1, \ldots, \alpha_{q-1}\}$ be non-zero elements in $\mathrm{GF}(q)$. Consider

$$\mathrm{GF}(q)' = \alpha \, \mathrm{GF}(q)^* = \{\alpha\alpha_1, \ldots, \alpha\alpha_{q-1}\}.$$

Note that if $\alpha\alpha_i, \alpha\alpha_j \in \mathrm{GF}(q)'$ and $\alpha\alpha_i = \alpha\alpha_j$, then $\alpha_i = \alpha_j$. Hence, the elements of $\mathrm{GF}(q)'$ are pairwise distinct. Also, $\alpha\alpha_i \neq 0$ for all $\alpha_i \in \mathrm{GF}(q)^*$. Hence, $\mathrm{GF}(q)^*$ is equal to $\mathrm{GF}(q)'$ up to ordering. Hence,

$$\prod_{i=1}^{q-1} \alpha_i = \prod_{i=1}^{q-1} \alpha\alpha_i = \alpha^{q-1} \prod_{i=1}^{q-1} \alpha_i.$$

Hence, $\alpha^{q-1} = 1$. Hence, $\alpha^q = \alpha$. □

Remark 2.3.17: Note that above theorem is generalization of Fermat's lil theorem. ◁

Definition 2.3.18: Let F be a field. We denote the ***set of non-zero elements (units) in*** F as F^*. i.e. $F^* = F \setminus \{0\}$. So, $\mathrm{GF}(q)^* = \mathrm{GF}(q) \setminus \{0\}$. ◁

Definition 2.3.19: Let $\alpha \in \mathrm{GF}(q)^*$. The ***order of*** α, denoted by $\mathrm{ord}(\alpha)$, is the smallest positive integer t that satisfies $\alpha^t = 1$. ◁

Example 2.3.20: There is only one element of order 1 in $\mathrm{GF}(q)$ and that is 1. ◁

Example 2.3.21: We want to find $\mathrm{ord}(x)$ where $x \in \mathrm{GF}(16) = \mathbb{Z}_2[x]/(x^4+x+1)$. One approach

we can take is to look x^i for all $i = 1, \ldots, 15$. We have

$$\begin{aligned}
x^1 &= x, & x^5 &= x^2 + x, & x^9 &= x^3 + 1, & x^{13} &= x^3 + x^2 + 1,\\
x^2 &= x^2, & x^6 &= x^3 + x^2, & x^{10} &= x^2 + x + 1, & x^{14} &= x^3 + 1,\\
x^3 &= x^3, & x^7 &= x^3 + x + 1, & x^{11} &= x^3 + x^2 + x, & x^{15} &= 1.\\
x^4 &= x + 1, & x^8 &= x^2 + 1, & x^{12} &= x^3 + x^2 + x + 1,
\end{aligned}$$

So we find $\text{ord}(x) = 15$ in GF(16). We also find that all of x^i are distinct for $1 \leq i \leq 15$. ◁

Theorem 2.3.22: Let $\alpha \in \text{GF}(q)^*$ with $\text{ord}(\alpha) = t$. Then, $\alpha^s = 1$ if and only if $t|\ s$.

Proof: Let $s \in \mathbb{Z}$. By division algorithm, we have $s = \ell t + r$. So, $\alpha^s = \alpha^{\ell t}\alpha r$. Hence, $\alpha^s = \alpha^{\ell t}\alpha r = (\alpha^t)^\ell \alpha^r = \alpha^r$. Thus, $\alpha^s = 1 \iff \alpha^r = 1 \iff r = 0 \iff t|\ s$. □

Corollary 2.3.23: $\text{ord}(\alpha)|q - 1$.

Remark 2.3.24: By the above theorem, to find $\text{ord}(x)$, we only needed to check x^1, x^3 and x^5 in Example 2.3.21. ◁

Definition 2.3.25: Let $\alpha \in \text{GF}(q)$. If $\text{ord}(\alpha) = q - 1$, then we say α is a ***generator*** of $\text{GF}(q)^*$ (or primitive element of $\text{GF}(q)$).

If α is a generator of $\text{GF}(q)^*$, then $\{\alpha^1, \ldots, \alpha^{q-1}\} = \text{GF}(q)^*$. So the powers of α *generate* $\text{GF}(q)^*$. ◁

The following material was covered in assignment 1.

Definition 2.3.26:

1. If $c_i \in C$ was sent, we denote the probability of IMLD making an incorrect decision as w_i.
2. The error probability of an $[n, M]$-code C is defined to be $P_C = \frac{1}{M}\sum_{i=1}^{M} w_i$. The error probability of a code varies depending on the probability distribution of source messages.

◁

Lemma 2.3.27: If $\alpha \in \text{GF}(q)^*$ has order α^t, then $\alpha^1, \ldots, \alpha^{t-1}$ are pairwise distinct.

Proof: Suppose, for contradiction, $\alpha^i = \alpha^j$ where $1 \leq i, j \leq t-1$ and i and j are distinct. WLOG, suppose $i > j$. Then, $i = j + k$ for some non-zero positive integer k where $1 \leq k \leq t - 2$. Then, $\alpha^i = \alpha^j\alpha^k$. Then, $\alpha^k = 1$ which is a contradiction since $\text{ord}(\alpha) = t > k$. □

Theorem 2.3.28: $\text{GF}(q)^*$ has at least one generator.

Proof: Optional reading posted on Learn. □

Example 2.3.29: To find a generator of $\text{GF}(8) = \mathbb{Z}_2[x]/(x^3 + x + 1)$, we want to find $\alpha \in \text{GF}(8)^*$ such that $\text{ord}(\alpha) = 7$. Since 7 is prime, then for any $\alpha \in \text{GF}(8)^*$, if $\alpha \neq 1$ then $\text{ord}(\alpha) = 7$. Hence, any non-identity $\alpha \in \text{GF}(8)$ is a generator. ◁

Chapter 3 – Linear Codes

Earlier we studied block codes and saw that block codes are just codes where all codewords have the same length. We used square brackets to denote block codes.

Remark 3.0.1: For $F = \mathrm{GF}(q)$, we denote the set of n-tuples over F as $V_n(F) = F \times \cdots \times F = F^n$. $V_n(F)$ is an n dimensional vector space and we have $|V_n(F)| = q^n$. ◁

Definition 3.0.2: A ***linear*** (n, k)***-code*** over F is a k dimensional subspace of the n-dimensional vector space $V_n(F)$. We use round brackets to denote linear codes and when we specify linear codes, we simply say "let C be an (n, k)-code" and assume the field F that C is associated to is a general finite field where $F = \mathrm{GF}(q)$. ◁

Recall 3.0.3: A subspace of a vector space V over F is a subset $S \subseteq V$ such that

1. $0 \in S$, so $S \neq \varnothing$, and
2. S is closed under addition and multiplication operations of F. i.e. for all $v_1, v_2 \in S$ and $\lambda \in F$ we have $v_1 + \lambda v_2 \in S$.

Note that a subspace $S \subseteq V$ is also a vector space. ◁

3.1 Properties of Linear Codes

Let C be an (n, k)-code over F. Let $v_1, \ldots, v_k$ be an ordered basis for C.

1. The codewords in C are precisely $m_1 v_1 + \cdots + m_k v_k$ where $m_i \in F$. So, $|C| = M = q^k$ since there are q choices for each m_k. The length of C is n and it has dimension of k.
2. The rate of C is $R = \frac{\log_q M}{n} = \frac{k}{n}$.
3. The distance of C is $d(C) = \min\{d(x, y) \mid x, y \in C \text{ and } x \neq y\}$.

Definition 3.1.1: The ***Hamming weight of*** $v \in V_n(F)$ is the number of non-zero coordinate positions in v, denoted by $w(v)$. The weight of C is $w(C) = \min\{w(c) \mid c \in C \text{ and } c \neq 0\}$. ◁

Theorem 3.1.2: If C is a linear code, then $d(C) = w(C)$.

Proof: We have

$$\begin{aligned} d(C) &= \min\{d(x, y) \mid x, y \in C \text{ and } x \neq y\}, \\ &= \min\{w(x - y) \mid x, y \in C \text{ and } x \neq y\}, \\ &= \min\{w(c) \mid c \in C \text{ and } c \neq 0\}, \\ &= w(C). \end{aligned}$$

□

Remark 3.1.3: Since $M = q^k$, there are q^k source messages. We will assume that the source messages are elements of $V_k(F)$. A natural encoding rule is, given $(m_1, \ldots, m_k) \in V_k(F)$, encode the message as $c = m_1 v_1 + \cdots + m_k v_k$.

Note that the encoding rule depends on the basis chosen for C and its order. ◁

Remark 3.1.4: If $m = (m_1, \ldots, m_k)$ and $v_1, \ldots, v_k$ is an ordered basis for C, then the encoding rule can be written as follows.

$$C = (m_1, \ldots, m_k) \begin{bmatrix} \text{—} & v_1 & \text{—} \\ \text{—} & v_2 & \text{—} \\ & \vdots & \\ \text{—} & v_k & \text{—} \end{bmatrix}_{k \times n} = mG.$$

In this course, we treat all vectors as row vectors. Here the matrix G, which is constructed from the basis vectors of C, is called the *generator matrix* (GM). ◁

Definition 3.1.5: Let C be a (n, k)-code. A ***generator matrix*** (GM) for C, denoted as G, is a $k \times n$ matrix where rows form a basis for C. ◁

Remark 3.1.6: An encoding rule for C with respect to G is $c = mG$. Performing elementary row operations on G gives a different generator matrix for the same code C. So, generator matrix is not unique. ◁

Example 3.1.7: Consider a binary $(5, 3)$-code C. Note that C has length 5 and dimension 3 and since C is a binary code, then $F = \text{GF}(2) = \mathbb{Z}_2$. We also have $|C| = q^k = 8$ and C has rate $\frac{3}{5}$. We can specify the code C by giving a basis as follows.

$$C = \langle 10010, 01011, 00101 \rangle = \langle v_1, v_2, v_3 \rangle.$$

To check v_1, v_2, v_3 are linearly independent we write them as rows. We have

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}_{3 \times 5}.$$

Since G is of the form $G_{3\times 5} = [I_3 \mid A]_{3\times 5}$, then v_1, v_2, v_3 are linearly independent. We also have that G is a generator matrix (GM) for C. Note that G is not the only GM for C . We have

m (Source)		c (codewords)		m (Source)		c (codewords)
000	$\to$	00000		011	$\to$	01110
001	$\to$	00101		101	$\to$	10111
010	$\to$	01011		110	$\to$	11001
100	$\to$	10010		111	$\to$	11100

Here we showed $mG = c \in C$ where G is defined as above. Note that different generator matrices (any matrix that is row equivalent to G) will have different encoding rules. We also see that $w(C) = 2 = d(C)$ and C has error detecting capability 1 and error correcting capability 0. ◁

Remark 3.1.8: If G a GM for an (n, k)-code and if it is in the form $[I_k \mid A]$, then it is trivial to go back to source messages from codewords. ◁

Definition 3.1.9: Let C be an (n, k)-code with GM G. If G is of the form $[I_k \mid A]$, then C is called a ***systematic code*** and G is said to be in standard form. ◁

Example 3.1.10: Let C be an $(6,3)$-code where $C = \langle 100011, 101010, 10010 \rangle$. We have

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

It is easy to see that since the second row of G is zero, then C is not systematic. We use ERO to convert G into RREF. We have

$$G \xrightarrow[R_2+R_3 \to R_3]{R_1+R_2 \to R_2} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Although C is not symmetric, if every codeword is permuted by moving the second bit to a new fourth bit, then we get a new code C' which is also linear and has the same length, dimension and distance as C. ◁

Definition 3.1.11: Let C be an (n,k)-code. If π is a permutation on $\{1,\ldots,n\}$ then $\pi(C)$ is an (n,k)-code and if two codes C and C' differ by a permutation on $\{1,\ldots,n\}$ then we say C and C' are ***equivalent codes***. Here $\{1,\ldots,n\}$ represent the coordinate positions of codewords.

If C and C' are equivalent codes, then $d(C) = d(C')$ since $w(C) = w(C')$. ◁

Theorem 3.1.12: Every linear code is equivalent to a systematic code.

Proof: Let C be an (n,k)-code. Let G be a GM for C where G is in RREF. Then, we can permute the columns of G to get a matrix $G_{\text{std.}}$ where $G_{\text{std.}} = [I_k \mid A]$ and $G_{\text{std.}}$ is in standard form. Then $G' = G_{\text{std.}}$ is a GM for code C' where C' is equivalent to C. □

3.2 Dual Codes

Definition 3.2.1: Let $x, y \in V_n(F)$. The (indefinite) inner (dot) product of x and y is defined as

$$x \cdot y = \sum_{i=1}^{n} x_i y_i \in F.$$

In this course we will simply refer this as dot product. If $x \cdot y = 0$, then we say x and y are ***orthogonal***. ◁

Theorem 3.2.2 (Properties of Inner Product)**:** For all vectors $x, y, z \in V_n(F)$ and scalars $\lambda \in F$ the following are true.

1. (Symmetry) $x \cdot y = y \cdot x$.
2. (Distributivity) $x \cdot (y + z) = x \cdot y + x \cdot z$.
3. (Associativity) $(\lambda x) \cdot y = \lambda(x \cdot y)$.
4. (Indefiniteness) $x \cdot x$ does not imply $x = 0$.

Proof: 1, 2 and 3 directly follow from commutativity and associativity of addition and multiplication in F. For 4, we see that for $(1,1) \in V_2(\mathbb{Z}_2)$, we have $(1,1) \cdot (1,1) = 0$. □

Definition 3.2.3: Let C be an (n,k)-code. The ***dual code of*** C is defined as

$$C^\perp = \{x \in V_n(F) \mid x \cdot c = 0, \forall\, c \in C\}. \qquad \triangleleft$$

Remark 3.2.4: For any linear code C, we see that $0 \in C^\perp$, so $C^\perp$ is non-empty. $\triangleleft$

Theorem 3.2.5: Let C be an (n,k)-code over F. $C^\perp$ be an $(n, n-k)$-code over F.

Proof: Let $C = \langle v_1, \ldots, v_k \rangle$. That is, $v_1, \ldots, v_k$ is a basis for C.

Claim 3.2.6*:* For any $x \in V_n(F)$, $x \in C^\perp$ if and only if $v_i \cdot x = 0$ for all $i = 1, \ldots, k$.

Proof: If $x \in C^\perp$ then for any $c \in C$ we have $x \cdot c = 0$. Since all $v_1, \ldots, v_k \in C$, then $v_i \cdot x = 0$ for $i = 1, \ldots, k$. For the converse, suppose $x \cdot v_i = 0$ for all $i = 1, \ldots, k$. Let $c \in C$. Then, $c = \lambda_1 v_1 + \cdots + \lambda_k + v_k$. Then,

$$x \cdot c = x\lambda_1 v_1 + \cdots + x\lambda_k + v_k = \lambda_1(x \cdot v_1) + \cdots + \lambda_k(x \cdot v_k) = 0.$$

Hence, $x \in C^\perp$. ■

Consider $G = (v_1, \ldots, v_k)^\top$. Then by the above claim $x \in C^\perp$ if and only if $xG^\top = 0$. So, $C^\perp$ is the nullspace of G. Hence, $C^\perp$ is an (n,k)-dimensional subspace of $V_n(F)$. □

Theorem 3.2.7: If C is a linear code, then $(C^\perp)^\perp = C$.

Proof: Let C be an (n,k)-code. Then $C^\perp$ is an $(n, n-k)$-code and $(C^\perp)^\perp$ is an (n,k)-code. Let c. Then, $c \cdot x = 0$ for all $x \in C^\perp$. So, $c \in (C^\perp)^\perp$. Hence, $C \subseteq (C^\perp)^\perp$. Suppose C is a code over $F = \mathrm{GF}(q)$. Then $C^\perp$ and $(C^\perp)^\perp$ are also codes over $F = \mathrm{GF}(q)$. Then $|C| = q^k = \left|(C^\perp)^\perp\right|$. Hence, $C = (C^\perp)^\perp$. □

Recall 3.2.8: For all $x_1, \ldots, x_k \in F$, $r_1, \ldots, r_m \in V_n(F)$ and $c_1, \ldots, c_n \in V_m(F)$ where $1 \le m, n \le k$ we have

$$(x_1, \ldots, x_m)\begin{bmatrix} \text{—} & r_1 & \text{—} \\ \text{—} & r_2 & \text{—} \\ & \vdots & \\ \text{—} & r_m & \text{—} \end{bmatrix}_{m\times n} = x_1 r_1 + \cdots + x_m r_m, \quad \text{and} \quad \begin{bmatrix} | & & | \\ c_1 & \ldots & c_n \\ | & & | \end{bmatrix}_{m\times n} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = c_1 x_1 + \cdots + c_n x_n.$$

$\triangleleft$

Theorem 3.2.9 (Constructing a GM for $C^\perp$)**:** Let C be an (n,k)-code over F with GM $G = [I_k \mid A]$. A GM is $C^\perp$ is $H = [-A^\top \mid I_{n-k}]$. Note that we have $G_{k\times n}$ and $A_{k\times(n-k)}$.

Proof: Clearly $\operatorname{Rank} H = n - k$, so H is a GM for some $(n, n-k)$-code $\overline{C}$. We also have

$$GH^\top = \begin{bmatrix} I_k \mid A \end{bmatrix}\left[\begin{array}{c} -A \\ \hline I_{n-k} \end{array}\right] = -A + A = 0.$$

Then, every row of H is orthogonal to every row of G. So, every vector in the row space of H is orthogonal to every vector in the row space of G. Hence, $\overline{C} \subseteq C^\perp$ and since $\dim \overline{C} = \dim C^\perp$, then $\overline{C} = C^\perp$. □

Example 3.2.10: Consider a $(5,2)$-code C over $\mathbb{Z}_3$ with GM $G = \begin{bmatrix} 2 & 0 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ where $|\mathbb{Z}_3| = q = 3$, $n = 5$ and $k = 2$. We have

$$C = \{ \underbrace{00000}_{(00)G}, \underbrace{20210}_{(10)G}, \underbrace{10120}_{(20)G}, \underbrace{11001}_{(01)G}, \underbrace{22002}_{(02)G}, \underbrace{01211}_{(11)G}, \underbrace{12212}_{(12)G}, \underbrace{21121}_{(21)G}, \underbrace{02122}_{(22)G} \}.$$

We have $w(C) = d(C) = 3$. So, C can correct at most $\lfloor \frac{d-1}{2} \rfloor = 1$ error. To find a GM for $C^\perp$, we convert G into $G_{\text{std.}}$ using ERO where $G_{\text{std.}}$ is in standard form as follows.

$$G \xrightarrow[2R_1+R_2 \to R_2]{2R_1 \to R_1} \begin{bmatrix} 1 & 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} I_2 \mid A \end{bmatrix}, \text{ where } A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 1 \end{bmatrix}.$$

So, by Theorem 3.2.9, we have

$$H = [-A^\top \mid I_3]_{3\times 5} = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & 1 \end{bmatrix},$$

where H is a GM for $C^\perp$ which is an $(5,3)$-code over $\mathbb{Z}_3$. ◁

Definition 3.2.11: A GM for $C^\perp$ is called a ***parity-check matrix*** (PCM) for C. ◁

3.3 Properties of Linear and Dual Codes

Remark 3.3.1: We make the following observations for an (n,k)-code C over F with GM G.

1. $C = \{mG \mid m \in V_k(F)\}$.
2. $C^\perp$ is an $(n, n-k)$-code over F.
3. $(C^\perp)^\perp = C$.
4. $G_{k\times n}$ is a GM for C and $H_{(n-k)\times n}$ is a GM for $C^\perp$.
5. $H_{(n-k)\times n}$ is a PCM for C and $G_{k\times n}$ is a PCM for $C^\perp$.
6. $C = \text{Row}(G)$ (C is rowspace of G) and $C^\perp = \text{Row}(H)$.
7. $C = \text{Null}(H)$ (C is nullspace of H) and $C^\perp = \text{Null}(G)$.
8. $x \in C$ if and only if $xH^\top = 0$ and $y \in C^\perp$ if and only if $yG^\top = 0$. ◁

Theorem 3.3.2: Let C be an (n,k)-code over F and let H be a PCM for C. Then, $d(C) \geq s$ if and only if every $s-1$ columns of H are linearly independent over F.

Proof: We will use contrapositive for both directions. Suppose there are $S-1$ columns of H that are linearly dependent over F. Let these columns be $h_1, \ldots, h_{s-1}$. Then, there exists scalars $c_1, \ldots, c_{s-1} \in F$ such that $c_1h_1 + \cdots + c_{s-1}h_{s-1} = 0$ where $c_1, \ldots, c_{s-1}$ are not all zero. Consider the codeword

$$c = (\underbrace{c_1, \ldots, c_{s-1}}_{s-1 \text{ times}}, \underbrace{0, \ldots, 0}_{n-s+1 \text{ times}}).$$

We have $Hc^\top = 0$. But then this shows there exists $c \in C$ such that $1 \leq w(c) \leq s-1$. Hence, $d(C) \leq s-1$. This proves the contrapositive of forward direction. Conversely, suppose $d(C) \leq s-1$. So, $w(C) \leq s-1$. Let $c \in C$ with $1 \leq w(c) \leq s-1$. WLOG, suppose $c_j = 0$ for all $j = s+1, \ldots, n$. So the non-zero entries in c are at the first $s-1$ coordinates. Let $h_1, \ldots, h_n$ be the columns of H. Since $c \in C$ then $Hc^\top = 0$. Hence, $c_1h_1 + \cdots + c_{s-1}h_{s-1} = 0$. Since $w(C) \geq 1$ by definition, then $h_1, \ldots, h_{s-1}$ are linearly dependent and this proves the contrapositive. □

Corollary 3.3.3: Let C be an (n, k)-code over F with PCM H. Then, $d(C)$ is the smallest number of columns H that are linearly dependent over F.

Example 3.3.4: Recall in Example 3.2.10 for a $(5, 2)$-code C over $\mathbb{Z}_3$, we found PCM of C as

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

We found that $d(C) = 3$. We will verify this by using the corollary above. Since H has no zero columns, then $d(C) > 1$. Since no two columns of H are scalar multiples of each other, then no two columns of H are linerly dependent, then $d(C) > 2$. Since we have

$$\begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = 2\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

then three of the columns of H are linearly dependent. Hence, $d(C) = 3$. ◁

Remark 3.3.5: Let C be a binary code with PCM H.

① $d(C) = 1$ if and only if H has a zero column. Note that this is true for any q-ary code with PCM H.

② $d(C) = 2$ if and only if the columns of H are non-zero and two columns of H are the same.

③ $d(C) = 3$ if and only if the columns of H are non-zero and distinct and there exists a column which is linear combination of two columns. ◁

Example 3.3.6: To construct a binary $(7, 4, 3)$-code C (that is, a binary $(7, 4)$-code of distance 3), we need to have a C's PCM must be of the form

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

Note that any column permutation of the above matrix is also a PCM for C. In this case, C is a *Hamming code* or order 3 over $\mathbb{Z}_2$. ◁

Definition 3.3.7: Let C be an $[n, M]$-code over A with $|A| = q$ and with $d(C) = d$. Then, by Assignment #1, we have

$$M \sum_{i=0}^{e} \binom{n}{i} (q-1)^i \leq q^n, \quad \text{where} \quad e = \left\lfloor \frac{d-1}{2} \right\rfloor \text{ (error correcting capability of } C).$$

This is known as ***sphere packing bound*** and it is a necessary condition for existence of a code but it's not a sufficient condition. We say C is ***perfect*** if

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

That is, in a perfect code every $r \in A^n$ is within distance e of some $c \in C$. This means for a perfect code C, IMLD is same as CMLD. ◁

Remark 3.3.8: For fixed n, q, d, a perfect code maximizes the rate, $R = \log_q M/n$. ◁

Example 3.3.9: The codes

① $C = \mathrm{GF}(q)^n = V_n(\mathrm{GF}(q))$,

② $C = \{\underbrace{0}_{\in\{0\}^n}, \underbrace{1}_{\in\{1\}^n}\}$ over $\mathbb{Z}_2$ where n is odd,

are perfect codes. The code in ② is called *binary replication code.* ◁

Theorem 3.3.10: Every perfect code has odd distance.

Proof: Exercise. ◁

Theorem 3.3.11 (Tietäväinen, 1973)**:** The only perfect codes are the following codes.

① $C = \mathrm{GF}(q)^n = V_n(\mathrm{GF}(q))$. This has $d = 1$.

② Binary replication code, $C = \{\underbrace{0}_{\in\{0\}^n}, \underbrace{1}_{\in\{1\}^n}\}$ of odd length. This has $d = n$.

③ The $(23, 12, 7)$-binary Golay code and all codes equivalent to it. We will cover this code in the following lectures.

④ The $(11, 6, 5)-$ternary Golay code and all codes equivalent to it. Ternary means over $\mathbb{Z}_3$. A GM for such a code is

$$G = \left[\begin{array}{c|ccccc} & 1 & 1 & 1 & 1 & 1 \\ & 0 & 1 & 2 & 2 & 1 \\ & 1 & 0 & 1 & 2 & 2 \\ I_6 & 2 & 1 & 0 & 1 & 2 \\ & 2 & 2 & 1 & 0 & 1 \\ & 1 & 2 & 2 & 1 & 0 \end{array}\right].$$

⑤ The Hamming codes and all codes of the same n, M, d parameters as them. These have $d = 3$.

Proof: Skipped, beyond the scope of this course. □

Definition 3.3.12: A ***Hamming code*** of order r over $\mathrm{GF}(q)$ is a linear code over $\mathrm{GF}(q)$ with $n = \frac{q^r-1}{q-1}$ and $k = n - r$ which has a PCM $H_{r\times n}$ whose columns are non-zero and no two columns are scalar multiplies of each other. ◁

Example 3.3.13: A Hamming code of order $r = 3$ over $\mathrm{GF}(2)$ is a $(7, 4, 3)$-binary code with PCM H where

$$H = \left[\begin{array}{c|cccc} & c_4 & c_5 & c_6 & c_7 \\ & 1 & 1 & 0 & 1 \\ I_3 & 1 & 0 & 1 & 1 \\ & 0 & 1 & 1 & 1 \end{array}\right].$$

It's clear this code has distance 3 since we have $c_4 + c_5 = c_6$. ◁

Example 3.3.14: A Hamming code of order $r = 3$ over GF(3) is a $(13, 10, 3)$- code with PCM H where

$$H = \left[\begin{array}{c|cccccccccc} & 1 & 1 & 0 & 1 & 2 & 2 & 0 & 2 & 2 & 1 \\ I_3 & 1 & 0 & 1 & 1 & 1 & 0 & 2 & 2 & 1 & 2 \\ & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 2 & 2 \end{array}\right]. \quad \triangleleft$$

Remark 3.3.15: We observe the following.

① For every non-zero vector $v \in V_r(\mathrm{GF}(q))$, exactly one scalar multiple of v must be a column of a PCM for the Hamming code of order r over GF(q).

② The dimension of Hamming codes is k since $\operatorname{rank} \mathrm{PCM} = r = n - k$ since $\lambda_i \mathbf{e}_i$ are columns of the PCM.

③ Hamming codes have distance 3.

④ Hamming codes are perfect. It easy to see this since we have $e = \left\lfloor \frac{d-1}{2} \right\rfloor = 1$ (so Hamming codes are 1-error correcting codes) and we have $M = q^k$ where $r = n - k$ and $n = \frac{q^r - 1}{q-1}$. So,

$$M \sum_{i=0}^{e} \binom{n}{i} (q-1)^i = q^{n-r}(1 + n(q-1)) = q^{n-r}(1 + q^r - 1) = q^n. \quad \triangleleft$$

Definition 3.3.16: If $c \in C$ is transmitted and $r \in V_n(F)$ is received, the vector e that satisfies $c + e = r$ is called the ***error vector***. Error vector is the vector that represents the error introduced to the code while transmission. ◁

Example 3.3.17: Consider $\mathrm{GF}(3) = \mathbb{Z}_3$. If $c = (120212)$ is sent and $r = (122102)$ is received, then $e = r - c = (002220)$. ◁

3.3.1 Decoding Linear Codes

3.3.1.1 Decoding Algorithms for 1-error Correcting Codes

Definition 3.3.18: Let $r \in V_n(\mathrm{GF}(q))$. The vector $s = Hr^\top$ is called the ***syndrome*** of r. ◁

Remark 3.3.19: We make the following remarks about syndromes.

① r and e have the same syndrome. So, if $e = 0$, then syndrome of e is $He^\top = 0$.

② If $w(e) = 1$, say $e = \mathbf{e}_i = (0, \ldots, 0, \alpha, 0, \ldots, 0)$, then $He^\top = \alpha h_i \neq 0$ where h_i is the i^{th} column of H. ◁

Remark 3.3.20: To perform decoding in 1-error correcting codes (e.g. Hamming codes), we will use an algorithm that follows the strategy below.

① Compute $s = Hr^\top$.

② If $w(s) = 0$, then accept r, otherwise

ⓐ Compare s with the columns of H. If $s = \alpha h_i$ where $\alpha \neq 0$, then take error vector as

$$e = \alpha \mathbf{e}_i = (0, \ldots, 0, \alpha, 0, \ldots, 0),$$

where α is in i^{th} position. Correct r to $c = r - e$.

③ Reject r.

Note that step ③ is skipped if C is a Hamming code since Hamming codes are perfect. ◁

Claim 3.3.21: If $w(e) \leq 1$, then the strategy above always makes the correct decision.

Proof: Exercise. □

Remark 3.3.22: If C is a Hamming code and $w(e) \geq 2$ then decoding algorithm above always makes the wrong decision. ◁

Example 3.3.23: Consider $(7, 4, 3)$-binary Hamming code with PCH H where

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

To decode $r = (0111110)$, we find $s = Hr^\top = [0\ 1\ 1]^\top$ which is the 6^{th} column of H. So, $e = (0000010)$. We see that $Hc^\top = 0$. ◁

3.3.1.2 General Decoding Problem for Binary Linear Codes

Remark 3.3.24: Consider an $(n-k) \times n$ matrix H (PCM of a code C with GM G) over GF(2) with $\operatorname{rank} H = n - k$ and $r \in V_n(\text{GF}(2))$. We want to find a vector $e \in V_n(\text{GF}(2))$ of minimum weight with $Hr^\top = He^\top$. This problem is NP-hard. ◁

3.3.1.3 Decoding Linear Codes in General

Definition 3.3.25: Let C be an (n, k)-code over $F = \text{GF}(q)$ with PCM H. We write $x \equiv y \pmod{C}$ where $x, y \in V_n(F)$ if $x - y \in C$ and say x is ***congruent*** to y. Since congruency is an equivalence relation we will write $x = y \mod C$. ◁

Definition 3.3.26: The equivalence class containing $x \in V_n(F)$ is called a ***coset*** of $V_n(F)$. This class is

$$\{y \in V_n(F) \mid y = x \bmod C\} = \{x + c \mid c \in C\}.$$

We write this class as $C + x = x + C$ and read it as the coset of C represented by x. ◁

Example 3.3.27: Consider the $(5, 2)$-binary code C with GM $G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$. So, $C = \{00000, 10111, 01110, 11001\}$. The cosets of C are

- $C + 00000 = \{00000, 10111, 01110, 11001\} = C + 10111 = C + 01110 = C + 11001$,
- $C + 10000 = \{10000, 00111, 11110, 01001\} = C + 00111 = C + 11110 = C + 01001$,
- $C + 01000 = \{01000, 11111, 00110, 10001\} = C + 11111 = C + 00110 = C + 10001$,

- $C + 00100 = \{00100, 10011, 01010, 11101\} = C + 10011 = C + 01010 = C + 11101$,
- $C + 00010 = \{00010, 10101, 01100, 11011\} = C + 10101 = C + 01100 = C + 11011$,
- $C + 00001 = \{00001, 10110, 01111, 11000\} = C + 10110 = C + 01111 = C + 11000$,
- $C + 00011 = \{00011, 10100, 01101, 11010\} = C + 10100 = C + 01101 = C + 11010$,
- $C + 11100 = \{11100, 01011, 10010, 00101\} = C + 01011 = C + 10010 = C + 00101$.

Since $d(C) = 3$, if $x \in V_n(F)$, then its syndrome is $s = Hr^\top \in V_{n-k} \in F$ where $|F| = q^{n-k}$. ◁

Remark 3.3.28: We make the following remarks.

1. Congruency is an equivalence relation. i.e. it's reflexive, symmetric and transitive.
2. The set of equivalence classes partitions $V_n(F)$.
3. $C + 0 = C$.
4. If $y \in C + x$, then $C + y = C + x$. This follows from the fact that congruency of codewords is and equivalence relation.
5. Every coset has cardinality of q^k.
6. There are $q^n/q^k = q^{n-k}$ cosets.

◁

Theorem 3.3.29: Let $x, y \in V_n(F)$. Then, $x = y \mod C$ if and only if $Hx^\top = Hy^\top$. So, cosets are characterized by their syndrome.

Proof: We have

$$\begin{aligned} x = y \mod C &\iff x - y \in C \\ &\iff H(x-y)^\top = 0 \\ &\iff (x-y)H^\top = 0 \\ &\iff xH^\top = yH^\top \iff Hx^\top = Hy^\top. \end{aligned}$$

□

Remark 3.3.30: If $c \in C$ is sent and $r \in V_n(F)$ is received, then we have $e = r - c \in V_n(F)$ and $Hr^\top = He^\top$. i.e. r and e have the same syndrome so they belong to the same coset of C.

CMLD: Given r, find a vector e of smallest weight in $C + r$ or, equivalently, find a vector e of smallest weight with the same syndrome as r then decode r to $c = r - e$.

IMLD: Given r, find the unique vector e of smallest weight in $C + r$ or, equivalently, find a vector e of smallest weight with the same syndrome as r. If no such e exists, reject r, otherwise decode r to $c = r - e$. ◁

3.3.1.4 Syndrome Decoding Algorithm

Definition 3.3.31: Given a PCM H for an (n, k)-code C over $F = \mathrm{GF}(q)$, a distinguished vector of smallest weight of a coset of C is called a ***coset leader*** of that coset. ◁

For syndrome decoding, we create a table of coset leaders and their syndromes. Given a received vector r, we consider the following algorithm.

Algorithm 3.3.32: Syndrome decoding algorithm.

1 **Given** r **do**
2 Compute $s = Hr^\top$.
3 $\ell \leftarrow$ the coset leader correspond to s.
4 Decode r to $c = r - \ell$.

Example 3.3.33: Recall Example 3.3.27. We had a $(5, 2)$-binary code with GM G and PCM H where

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}_{2\times 5}, \quad \text{and} \quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{3\times 5}.$$

Since $q^{n-k} = 2^{5-2} = 2^3 = 8$, we have 8 cosets and the table of coset leaders and syndromes is as follows.

Coset Leaders		Syndromes
00000	———	000
10000	———	111
01000	———	110
00100	———	100
00010	———	010
00001	———	001
$00011^{\dagger_1}$	———	011
$10010^{\dagger_2}$	———	101

Note that the cosets that correspond to leaders with $\dagger_1$ and $\dagger_2$ are

$$C + 00011 = \{00011, 10100, 01101, 11010\},$$
$$C + 10010 = \{11100, 01011, 10010, 00101\}.$$

We see that in both of these cosets the vector with the minimum weight is not unique. By Theorem 3.3.29, any vector in these cosets give the correct syndrome. So if we picked different coset leaders, say 10100 for $\dagger_1$ and 00101 for $\dagger_2$, then we would get the same corresponding syndromes, say $s_{\dagger_1}$ and $s_{\dagger_2}$. In this case, if we receive any vector r such that $Hr^\top = s_{\dagger_i}$ for $i = 1, 2$, then the coset leader-syndrome table determines how we decode r. Indeed if this is the case, if the algorithm finds that error vector is a coset leader, then the received word will be decoded to the codeword that was in fact transmitted. i.e. all error vectors that are not coset leaders in the syndrome decoding algorithm will be handled incorrectly.

Consider the table above and suppose $r = 10111$ is received. We find $s = Hr^\top = [0\ 0\ 0]^\top$. So, coset leader is $\ell = 00000$ and $c = r - e = 10111$. ◁

Chapter 4 – The Binary Golay Code

Remark 4.0.1: For a binary (n, k)-code C, the syndrome table has size $2^{n-k} \times n$, which is exponentially large. We want to design decoding algorithms which require very little space. e.g. use only the PCH H which is $(n-k)n$ bits. ◁

4.0.1 The Binary Golay Code C_{23} (1949)

Let $\hat{B}$ be the matrix below.

$$\hat{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ & & & & & \vdots & & & & & \\ & & & & & \vdots & & & & & \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{12\times 11} \Bigg\} \text{rows 3 to 12 are left cyclic shifts of second row.}$$

So we have $B = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{12} \end{bmatrix}$, where $r_1 = [1 \dots 1]_{1\times 11}$ and $r_2 = [1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0]$ and $r_i = \sigma_{i-2}(r_2)$ for $i = 3, \dots 12$ where $\sigma_n(r)$ is imposes n left shifts on r. Let $\hat{G} = [I_{12}\ \hat{B}]_{12\times 23}$. So, $\hat{G}$ is a GM for a $(23, 12)$-binary code called C_{23}. We see that $d(C_{23}) = 7$ and since $\left\lfloor \frac{7-1}{2} \right\rfloor = 3$, then C_{23} is a 3-error correcting code. We also have that C_{23} is perfect since

$$2^{12}\left[\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}\right] = 2^{23}.$$

4.0.2 The Extended Golay Code C_{24}

We want to add a parity bit to each row in $\hat{G}$. Since the first row of $\hat{B}$ has all 1's and since $w(r_i) = 6$ for all $i = 2, \dots, 12$, we add a column c to $\hat{B}$ on the right where

$$c = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{12\times 1}.$$

So we have $B = [c \mid \hat{B}]$ and $G = [I_{12} \mid B_{12}]$. So, G is a GM for code C_{24}.

Definition 4.0.2: If a code C satisfies $C \subseteq C^{\perp}$, then C is called ***self-orthogonal*** or ***self-dual*** code. ◁

Remark 4.0.3: We make the following remarks.

1. C_{24} is a $(24, 12, 8)$-binary code.
2. $GG^\top = 0$.
3. $C_{24} \subseteq C_{24}^\perp$ and since $\dim C_{24} = 12 = \dim C_{24}^\perp$, then $C_{24} = C_{24}^\perp$.
4. B is symmetric.
5. A PCM for C_{24} is $H = [B^\top \mid I_{12}] = [B \mid I_{12}]$.
6. Since $C_{24} = C_{24}^\perp$, then $H = [B \mid I_{12}]$ is both PCM and GM for C and $G = [I_{12} \mid B]$ is also both GM and PCM for $C_{24}^\perp$. ◁

4.0.2.1 Decoding Algorithm for C_{24}

Remark 4.0.4: The syndrome table for C_{24} has size $2^{12} \times 24 \approxeq 96\,000$ bits. The general decoding strategy is as follows.

- Compute syndrome of r. That is, find $s = Hr^\top$.
- Find a vector e of weight at most 3 that has the same syndrome as r.
- If no such vector e exists, then reject r, otherwise decode r to $c = r - e$. ◁

Theorem 4.0.5: Let C be an (n, k, d)-code over $\mathrm{GF}(q)$. Let $x \in V_n(\mathrm{GF}(q))$ with $w(x) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. Then, x is the unique vector of minimum weight in the coset of C that contains x. i.e. x is the only coset leader in $C + x$.

Proof: Suppose, for contradiction, there exists a vector y in the same coset of C as x (so, $y \in C+x$) with $x \neq y$ and

$$w(y) \leq w(x) \leq \left\lfloor \frac{d-1}{2} \right\rfloor.$$

We have $y - x \neq 0$ and $x = y \mod C$. So, $x - y \in C$. But this gives us

$$w(x-y) = w(x + (-y)) \leq w(x) + w(-y) = w(x) + w(y) \leq \left\lfloor \frac{d-1}{2} \right\rfloor + \left\lfloor \frac{d-1}{2} \right\rfloor \leq d,$$

which is a contradiction since $d(C) = d$. □

We use the following approach to decode C_{24}. Note that the following approach is different from the one in textbook.

Remark 4.0.6: Let $r = (x, y)$ and $e = (e_1, e_2)$ where $x, y, e_1, e_2 \in V_{12}(\mathbb{Z}_2)$. i.e. x, y, e_1, e_2 are 12 bits. We consider 5 cases which are not mutually exclusive.

- **A** $w(e_1) = 0 = w(e_2)$.
- **B** $1 \leq w(e_1) \leq 3$ and $w(e_2) = 0$.
- **C** $1 \leq w(e_1) \leq 2$ and $w(e_2) = 1$.
- **D** $w(e_1) = 0$ and $1 \leq w(e_2) \leq 3$.
- **E** $w(e_1) = 1$ and $1 \leq w(e_2) \leq 2$.

We have

$$s_1 = [I_{12} \mid B]r^\top = [I_{12} \mid B]e^\top = [I_{12} \mid B]\begin{bmatrix} e_1^\top \\ e_2^\top \end{bmatrix} = e_1^\top + Be_2^\top,$$
$$s_2 = [B \mid I_{12}]r^\top = [B \mid I_{12}]e^\top = [B \mid I_{12}]\begin{bmatrix} e_1^\top \\ e_2^\top \end{bmatrix} = Be_1^\top + e_2^\top.$$

Suppose that $r = (x, y)$ is received and $e = (e_1, e_2)$ where $x, y, e_1, e_2 \in V_{12}(\mathbb{Z}_2)$. i.e. x, y, e_1, e_2 are 12 bits. Note that since C_{24} is a binary code, correcting x in position j means flipping the j^{th} bit of x.

1. Find s_1. If $s_1 = 0$, then accept r and STOP.
2. If $w(s_1) \leq 3$, then correct x in the positions corresponding to the 1's in s_1 and STOP.
3. Compare s_1 to the columns (or rows) of B. If any row of B, say column i, differs in 1 position, say position j, or 2 positions, say positions j and k, from s_1, then correct r as follows and STOP.
   - Correct x in position j or in positions j and k.
   - Correct y in position i.
4. Find s_2.
5. If $w(s_2) \leq 3$, then correct y in the positions corresponding to the 1's in s_2 and STOP.
6. Compare s_2 to the columns (or rows) of B. If any row of B, say column i, differs in 1 position, say position j, or 2 positions, say positions j and k, from s_1, then correct r as follows and STOP.
   - Correct y in position j or in positions j and k.
   - Correct x in position i.
7. Reject r.

◁

In Python, we have the following code http://www.student.math.uwaterloo.ca/~c2kent/LectureNotes/co331-1201/decodingC24python.pdf.

Example 4.0.7: If $r = (1000\ 1000\ 0000\ 1001\ 0001\ 1101)$ is received, then we have

$$\begin{aligned} x &= (1000\ 1000\ 0000), \\ y &= (1001\ 0001\ 1101), \end{aligned} \quad \text{and} \quad s_1 = [0100\ 1000\ 0000]^\top.$$

Since $w(s_1) \leq 3$, we set $e = (e_1, e_2) = (s_1^\top, 0)$ and decode r to

$$c = r - e = (1100\ 0000\ 0000\ 1001\ 0001\ 1101).$$

Since we have $Hc^\top = Gc^\top = 0$, then $c \in C_{24}$.

If $r = (1000\ 0010\ 0000\ 1000\ 1101\ 0010)$ is received, then we have

$$\begin{aligned} x &= (1000\ 0010\ 0000), \\ y &= (1000\ 1101\ 0010), \end{aligned} \quad \text{and} \quad s_1 = [1011\ 1110\ 1011]^\top.$$

Since $w(s_1) > 3$, we compare s_1 with the rows of B. Since s_1 differs in positions 6 and 7 from row 4 of B, we set $e = (e_1, e_2) = (0000\ 0110\ 0000\ 0001\ 0000\ 0000)$, and decode r to

$$c = r - e = (1000\ 0100\ 0000\ 1001\ 1101\ 0010).$$

$Hc^\top = Gc^\top = 0$, then $c \in C_{24}$. ◁

Remark 4.0.8: We make the following remarks about the algorithm.

① If $w(e) \leq 3$, then the algorithm makes the correct decision.

② Algorithm requires no storage (it doesn't require any memory) since we have

$$s_1 = [I_{12} \mid B]r^\top = [I_{12} \mid B]\begin{bmatrix} x_{12\times 1} \\ y_{12\times 1} \end{bmatrix} = x + By,$$

which is easily calculated, similarly for s_2.

③ Algorithm is very simple and efficient, so it's good for hardware. ◁

4.0.2.2 Reliability of C_{24}

C_{24} is a BSC. We will compare C_{24} with other codes to see how reliable it is. Note that triplication code is a replication code when $n = 3$, which is the code in third row in Example 0.1.1. We recall the following definitions that we made in Definition 2.3.26.

- p : symbol error probability.
- $C = \{c_1, \ldots, c_M\}$ where $c_i \in C$ are codewords in code C for $i = 1, \ldots, M$.
- w_i : probability that decoding algorithm makes an incorrect decision if c_i is sent.
- P_C : error probability of C. We have

$$P_C = \frac{1}{M}\sum_{i=1}^{M} w_i.$$

- $1 - P_C$: Reliability of C.

We have

p	① $(1-p)^{12}$	② $1 - P_{C_{24}}$	③ $1 - P_T$	④ $1 - P_H$
0.1	0.282429	0.7857379	0.7112056	0.5490430
0.01	0.8863848	0.99990946	0.9964298	0.9903702
0.001	0.9886657	0.9999999895	0.99996402	0.9998959
Rate:	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{11}{15} \approx 0.73$

where

① If no source is used, then the reliability for 12-bit messages is $(1-p)^{12}$.

② For C_{24} we have

$$w_i = \left[(1-p)^{24} + \binom{24}{1}p(1-p)^{23} + \binom{24}{2}p^2(1-p)22 + \binom{24}{3}(1-p)^{21}\right].$$

Hence,

$$P_{C_{24}} = \frac{1}{2^{12}}\sum_{i=1}^{2^{12}} w_i = w_i, \text{ since all } w_i \text{ are the same.}$$

③ Here T is triplication code. We have

$$1 - P_T = [(1-p)^3 + 3p(1-p)^2]^{12}.$$

④ Here H is Hamming code. The closest Hamming code we have to C_{24} is the $(15, 11)$-binary hamming code. We have

$$1 - P_H = (1-p)^{15} + 15p(1-p)^{14}.$$

Chapter 5 – Cyclic Codes

5.1 The Association Between $S \subseteq V_n(F)$ and $R = F[x]/(x^n - 1)$

Definition 5.1.1: A ***cyclic subspace*** S of $V_n(F)$ is a subspace such that if $(a_0, \ldots, a_{n-1}) \in S$ then $(a_{n-1}, a_0, \ldots, a_{n-2}) \in S$.

A ***cyclic code*** is a subspace of $V_n(F)$. ◁

Remark 5.1.2: Let $R = F[x]/(x^n-1)$ be a ring where F is a field and $F[x]$ is the set of polynomials over F. We showed that R is a commutative finite ring. We can associate $a = (a_0, \ldots, a_{n-1}) \in S$ with elements in R in a natural way as follows.

$$(a_0, a_1, \ldots, a_{n-1}) \quad \longleftrightarrow \quad a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in R.$$

This association conserves addition and scalar multiplication since

$$\begin{aligned} a + b \quad &\longleftrightarrow \quad a(x) + b(x), \\ \lambda a \quad &\longleftrightarrow \quad \lambda a(x). \end{aligned}$$

We chose $R = F[x]/(x^n - 1)$ since $x^n = 1 \mod (x^n - 1)$, the association we have gives

$$xa(x) = a_0x + a_1x^2 + \cdots + a_{n-1}x^n = a_{n-1} + \cdots + a_{n-2}x^{n-1} \mod (x^n - 1) \longleftrightarrow (a_{n-1}, a_0, \ldots, a_{n-2}).$$

So, multiplying a polynomial in R by x corresponds to a (right) cyclic shift of the associated vector.

We define multiplication as a binary operation $\cdot : V_n(F) \times V_n(F) \to V_n(F)$ by

$$a \cdot b \quad \longleftrightarrow \quad a(x) \cdot b(x) \mod (x^n - 1), \quad \forall\, a, b \in V_n(F). \qquad \triangleleft$$

5.1.1 Ideals of $R = F[x]/(x^n - 1)$

Definition 5.1.3: Let R be a commutative finite ring. A non-empty subset $I \subseteq R$ is called an ***ideal*** of R if

1. for all $a, b \in I$, $a + b \in I$,
2. for all $a \in I$ and $c \in I$, $ac \in I$. ◁

Example 5.1.4: For any ring R, 0 and R are ideals of R. These are called *trivial ideals.* ◁

Theorem 5.1.5: Let $S \subseteq V_n(F)$ be non-empty and let I be the set of associated polynomials as described above. Then, S is a cyclic subspace of the vector space $V_n(F)$ if and only if I is an ideal of $R = F[x]/(x^n - 1)$.

Proof: Suppose S is a cyclic subspace of $V_n(F)$. Since S is non-empty and closed under vector addition, then so is I. Let $a(x) \in I$ and $b(x) = b_0 + b_1x + \cdots + b_{n-1}x^{n-1} \in R$. Then, $xa(x) \in I$ since S is a cyclic subspace. Then, $x^i a(x) \in I$ for all $i = 0, \ldots, n-1$. Since for all scalars $b_i \in F$ we also

have $b_i \in R$, then $b_i x^i a(x) \in F$ for all $i = 0, \ldots, n-1$since S is closed under scalar multiplication. Finally,

$$a(x)b(x) = a(x)(b_0 + b_1 x + \cdots + b_{n-1}x^{n-1}) = b_0 a(x) + b_1 a(x)x + \cdots + b_{n-1}a(x)x^{n-1} \in I$$

since I is closed under addition and scalar multiplication. Hence, I is an ideal. Conversely, suppose I is an ideal of R. Since I is closed under addition then so is S. For any scalars $k \in F$ we also have $k \in F[x]$. So, $k \in R$ are constants. Since R is closed under multiplication by all polynomials in R, then it's also closed under multiplication by constant polynomials. Hence, S is closed under scalar multiplication. Finally, since I is closed under multiplication by x, then for any $s \in S$, we have $xs \in S$. So, S is closed under (right) cyclic shifts. Hence, S is a cyclic subspace. □

Remark 5.1.6: This theorem shows we have a one-to-one correspondence between cyclic subspaces of $V_n(F)$ and the ideals of $R = F[x]/(x^n - 1)$. ◁

Definition 5.1.7: Let R be a ring. If the set $\langle g(x)\rangle = \{g(x) \cdot a(x) \mid a(x) \in R\}$, is an ideal of R, then it's called *the ideal generated by* $g(x)$.

Let I be an ideal of R. If there exists $g(x) \in I$ such that $I = \langle g(x)\rangle$ then I is called a ***principal ideal***. If every ideal of R is principal, then R is called a ***principal ideal ring***. ◁

Theorem 5.1.8: The ring $R = F[x]/(x^n - 1)$ is a principal ideal ring.

Proof: If $I = \{0\}$ then $I = \langle 0\rangle$ is principal. Suppose $I \neq \langle 0\rangle$. Let $g(x)$ be a non-zero polynomial of smallest degree in I. Let $a(x) \in I$. By division algorithm we have

$$a(x) = \ell(x)g(x) + r(x)$$

where $\ell(x), r(x) \in F[x]$ and $\deg r < \deg g$. Then, there exists $[\ell(x)] \in R$ such that $\ell(x) \in [\ell(x)]$. Since I is closed under addition and multiplication by polynomials in R, then $\ell(x)g(x) \in I$. Hence,

$$a(x) - \ell(x)g(x) = r(x) \in I.$$

Since $\deg r < \deg g$, then we must have $r(x) = 0$. Hence, for any $a(x) \in I$, we have $a(x) = \ell(x)g(x)$. Hence, $I = \langle g(x)\rangle$. Hence, R is a principal ideal ring. □

Remark 5.1.9: In the proof of Theorem 5.1.8, if $I \neq \{0\}$ then we took $g(x) = a$ as a non-zero polynomial of smallest degree in I. Note that we can take $g(x)$ to be monic (leading coefficient is 1). If $g(x)$ is not monic, say $g(x) = g_\ell x^\ell + g_{\ell-1}x^{\ell-1} + \cdots + g_1 x^1 + g_0$ where g_ℓ is non-zero and $g_\ell \neq 1$. Since g_ℓ^{-1} exists in F, then the constant polynomial g_ℓ^{-1} resides in $F[x]/(x^n - 1)$. Hence, $g_\ell^{-1}g \in I$ where $g_\ell^{-1}g$ is monic. This process is called *making* $g(x)$ *monic*. ◁

Definition 5.1.10: Let I be an ideal in $F[x]/(x^n-1)$. If $I = \{0\}$, then ***the generator polynomial of*** I is defined to be x^n-1 since $x^n-1 = 0 \bmod x^n-1$. If $I \neq \{0\}$, then the unique monic polynomial with least degree in I is called ***the generator polynomial of*** I. ◁

Theorem 5.1.11: Let I be a non-zero ideal in $R = F[x]/(x^n - 1)$. Then,

1. there is a unique monic polynomial $g(x)$ of smallest degree in I,
2. $g(x)|x^n - 1$.

Proof: We first show uniqueness. Suppose $g(x), h(x)$ are two monic polynomials in I with same smallest degree where $\deg g = \ell = \deg h$. Denote $g(x) - h(x) = r(x)$. We have $\deg r < \ell$ and $r \in I$.

If r is non-zero, then we can make r monic and have a monic polynomial in I with degree less than ℓ which is a contradiction. Hence, r must be the zero polynomial. Hence, $g = h$. For the second part of the theorem, we can write $x^n - 1 = \ell(x)g(x) + r(x)$ where $\ell(x), r(x) \in F[x]$ and $\deg r < \deg g$. So,

$$0 = \ell g + r \mod x^n - 1.$$

This gives us

$$r = -lg \mod x^n - 1.$$

Hence, there exists $[\ell] \in R = F[x]/(x^n - 1)$ such that $\ell = [\ell] \mod x^n - 1$. Since $g(x)$ generates I, then $\langle g(x) \rangle = I$. So, $r(x) = -\ell(x)g(x) \in I$. Hence, $\deg r \leq \deg g$ which means $r(x)$ is the zero polynomial. Hence, $g(x)|x^n - 1$. □

Theorem 5.1.12: Let $h(x)$ be a monic divisor of $x^n - 1$ in $F[x]$. Then, the unique generator polynomial of $\langle h(x) \rangle$ is $h(x)$.

Proof: If $h(x) = x^n - 1$, then $I = \{0\}$ and by definition, its generator polynomial is $x^n - 1$. If $\deg h < n$, then $I \neq \{0\}$. Let g be the monic polynomial with smallest degree in $I = \langle h \rangle$. Since $h(x)$ is a generator of $\langle h \rangle$ and since $g \in \langle h \rangle$, then $g = ah \mod x^n - 1$ for some polynomial $a \in R = F[x]/(x^n - 1)$. Then,

$$g(x) = a(x)h(x) + \ell(x)(x^n - 1),$$

for some $\ell(x) \in F[x]$. Since $h|x^n - 1$, then $h|ah + \ell(x^n - 1)$. Hence $h|g$. Since $\deg g \leq \deg h$ by construction, and since both h and g are monic, then we must have $h = g$. □

Corollary 5.1.13: There is a one-to-one correspondence (bijection) between monic divisors of $x^n - 1$ in $F[x]$ and ideals in $R = F[x]/(x^n - 1)$.

Hence, there is a one-to-one correspondence (bijection) between monic divisors of $x^n - 1$ in $F[x]$ and cyclic subspaces of $V_n(F)$.

Example 5.1.14: Find all cyclic subspaces of $V_3(\mathbb{Z}_2)$. We have $n = 3$. The complete factorization of $x^3 - 1$ over $\mathbb{Z}_2$ is

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

Hence, we have the monic divisors of $x^3 - 1$ as

$$\begin{aligned} g_1 &= 1, \\ g_2 &= x + 1 = 1 + x, \\ g_3 &= x^2 + x + 1 = 1 + x + x^2, \\ g_4 &= x^3 - 1 = 1 + x^3 = 0. \end{aligned}$$

Note that g_1 and g_4 correspond to trivial ideals. By the one-to-one association we made in the beginning of this chapter, we have the following correspondence.

$\langle g_1(x) \rangle = R = F[x]/(x^3 - 1) = \{000, 001, \ldots, 111\} = S_1$ (since $\forall\ f \in R,$ we have $g_1 f = f \in I$),
$\langle g_2(x) \rangle = \{000, 110, 011, 101\} = S_2$,
$\langle g_3(x) \rangle = \{000, 111\} = S_3$,
$\langle g_4(x) \rangle = \{000\} = S_4$ (since $g_4 f = 0$ for any $f \in R$).

Note that it is easy to verify $\langle g_2(x)\rangle$ and $\langle g_3(x)\rangle$ correspond to S_2 and S_3 respectively. They clearly contain additive identity of R, the zero polynomial, which corresponds to 000 in $V_3(\mathbb{Z}_2)$. They also contain their generators, $g_2(x)$ and $g_3(x)$. Hence, their corresponding subspaces contain the vector representation of their generator polynomials. Since cyclic subspaces are closed under cyclic shifts, then they also contain the cyclic permutation of each their elements. Since the generators of the ideals $\langle g_2(x)\rangle$ and $\langle g_3(x)\rangle$ are the unique monic polynomials with smallest degree, then their corresponding cyclic subspaces cannot contain representations of lesser degree polynomials. This means, for $\langle g_3(x)\rangle$, S_3 only contains 000 and 111. ◁

Remark 5.1.15: In the beginning of this chapter we saw that there is a one-to-one association between cyclic codes (subspaces of $V_n(F)$) and the ring $R = F[x]/(x^n - 1)$. We denoted this association with $\leftrightarrow$. This association extends as follows, some of which we already proved in previous lectures.

$$
\begin{array}{lcl}
V_n(F) & \longleftrightarrow & R = F[x]/(x^n-1), \\
a = (a_0, a_1, \ldots, a_{n-1}) \in V_n(F) & \longleftrightarrow & a_0 + a_1x^1 + \cdots + a_{n-1}x^{n-1} \in R = F[x]/(x^n-1), \\
C, \text{ (cyclic subspace) with } \dim C = k & \longleftrightarrow & I, \text{ (ideal in } R), \\
 & \longleftrightarrow & g(x), \text{ (monic div. of } x^n-1) \text{ with } \deg g = n-k, \\
\text{if } G \text{ is a GM for } C \text{ in terms of } g(x), & & \\
\text{Encoding: } mG & \longleftrightarrow & m(x)g(x), \\
C^\perp, \text{ (dual code of } C \text{ which is also cyclic)} & \longleftrightarrow & h^*(x), \text{ (generator polynomial of } C^\perp), \\
\text{PCM } H \text{ for } C & \longleftrightarrow & s(x) = r(x) \bmod g(x).
\end{array}
$$

To find $h^*(x)$, we find $h(x) = (x^n-1)/g(x)$ where $\deg h = k$ We then find the reciprocal polynomial $h_R(x)$ if $h(x)$. Finally, we make $h_R(x)$ monic and find $h^*(x)$. ◁

5.2 Constructing Cyclic Codes

Remark 5.2.1: To find the distance of a cyclic code C, we consider BCH (Bose–Chaudhuri–Hocquenghem) codes. In this case, we will specially select $g(x)$ to have a lower bound on distance d of C. ◁

Lemma 5.2.2: Let $g(x)$ be a monic divisor of $x^n - 1$ with degree $n - k$ in $F[x]$. Then

$$\langle g(x)\rangle = \{g(x)a(x) \mid a(x) \in R = F[x]/(x^n-1)\} = \{g(x)\overline{a}(x) \mid \deg \overline{a}(x) \le k-1\}.$$

Proof: Let $h(x) = g(x)a(x) \bmod (x^n - 1)$ for some $a(x)$ with $\deg a(x) < n$. So, $h(x) - g(x)a(x) = \ell(x)x^n - 1$ for some $\ell(x) \in F[x]$. Hence, $h(x) = g(x)a(x) + \ell(x)x^n - 1$. Since $g(x)|x^n - 1$, then $g(x)|h(x)$. Hence, $h(x) = g(x)\overline{a}(x)$ for some $\overline{a}(x) \in F[x]$ with degree $\deg \overline{a}(x) \le k - 1$. □

Theorem 5.2.3: Let g be a monic divisor of $x^n - 1$ with degree $n - k$ in $F[x]$. Then, the cyclic code C generated by $g(x)$ has dimension k.

Proof: We will show that $B = \{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ is a basis of C. We first show B is linearly independent. Consider the linear combination of polynomials in B with constants λ_i from F.

$$\lambda_0 g(x) + \lambda_1 xg(x) + \cdots + \lambda_{k-1}x^{k-1}g(x).$$

We have $\deg(\lambda_{k-1}x^{k-1}g(x)) = n-k+k-1 = n-1$ and

$$\deg(\lambda_i x^i g(x)) = \deg(\lambda_{i+1}x^{i+1}g(x)) - 1 = n-k+i, \text{ for } i = 0, \ldots, k-2.$$

Hence, if $\lambda_0 g(x) + \lambda_1 x g(x) + \cdots + \lambda_i x^i g(x) = 0$, we must have $\lambda_i = 0$. Then, if

$$\lambda_0 g(x) + \lambda_1 x g(x) + \cdots + \lambda_{k-1}x^{k-1}g(x) = 0,$$

it follows that $\lambda_i = 0$ for all $i = 0, \ldots, k-1$. Hence, B is linearly independent. We now show B spans C. Let $h(x) \in \langle g(x)\rangle$. By the above lemma, we can write $h(x) = g(x)a(x)$ for some $a(x) \in F[x]$ where $\deg g(x) = n-k$ and $\deg a(x) \leq k-1$. Let

$$a(x) = \sum_{i=0}^{k-1} a_i x_i, \text{ where } a_i \in F.$$

Then, we have

$$h(x) = g(x)a(x) = g(x)\sum_{i=0}^{k-1} a_i x_i = \sum_{i=0}^{k-1} a_i x_i g(x), \text{ where } a_i \in F.$$

Hence, $h(x)$ can be expressed as linear combination of polynomials in B. Since this is true for any $h(x) \in \langle g(x)\rangle$, then B spans $\langle g(x)\rangle$ and hence, B spans C. Since $|B| = k$, then $\dim C = k$. □

Remark 5.2.4: Using the basis B for a cyclic code C generated by $g(x)$, we can find a GM G for C as writing the polynomials in B as rows in G. By using the vector $\leftrightarrow$ polynomial association, we have

$$\begin{aligned}
g(x) &= g_0 + g_1 x + \cdots + g_{n-k-1}x^{n-k-1} + x^{n-k} &&\longleftrightarrow (g_0, g_1, \ldots, g_{n-k-1}, 1),\\
xg(x) &= g_0 x + g_1 x^2 + \cdots + g_{n-k-1}x^{n-k} + x^{n-k-1} &&\longleftrightarrow (0, g_0, g_1, \ldots, g_{n-k-1}, 1),\\
&\vdots &&\qquad\vdots\\
x^{k-1}g(x) &= g_0 x^{k-1} + g_1 x^k + \cdots + g_{n-k-1}x^{n-2} + x^{n-1} &&\longleftrightarrow (\underbrace{0, \ldots, 0}_{k-1 \text{ times}}, g_0, g_1, \ldots, g_{n-k-1}, 1).
\end{aligned}$$

Hence, we have

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix} \longleftrightarrow \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k-1} & 1 & 0 \\ 0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k-1} & 1 \end{bmatrix}_{k\times n}.$$

Note that this G is non-systematic. Encoding in C is as follows. Let $c \in C$ be a codeword. Then,

$$\begin{aligned}
c = mG &= (m_0, m_1, \ldots, m_{k-1}) \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}\\
&= m_0 g(x) + m_1 x g(x) + \cdots + m_{k-1}x^{k-1}g(x)\\
&= (m_0 + m_1 x + \cdots + m_{k-1}x^{k-1})g(x)\\
&= m(x)g(x).
\end{aligned}$$

◁

Example 5.2.5: To construct a cyclic $(7,4)$-code C over $\mathbb{Z}_2$, we need a GM G for C that satisfies above. So, we need a monic divisor of x^7-1 with degree $7-4=3$ in $Z_2[x]$. From table 3 in page 157 of the textbook, we find

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

Both $(1+x+x^3)$ and $(1+x^2+x^3)$ are monic divisors of x^7-1 so we can pick either of them. We pick $(1+x+x^3)$. Then, $\langle g(x)\rangle$ is a $(7,4)$ cyclic code C over $\mathbb{Z}_2$. Since

$$1+x+x^3 \longleftrightarrow (1101)$$

then, a GM G for C is

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

So, if we want to encode $m=(1011)$, we have

$$mG=(1011)\begin{bmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\0&0&1&1&0&1&0\\0&0&0&1&1&0&1\end{bmatrix}=(1111111).$$

Equivalently, we have $m=(1011)\longleftrightarrow 1+x^2+x^3$. So,

$$m(x)g(x)=(1+x^2+x^3)(1+x+x^3)=1+x+x^2+x^3+x^4+x^5+x^6,$$

which corresponds to (111111) as expected. ◁

5.3 Dual Code of a Cyclic Code

Remark 5.3.1: Let C be an (n,k)-cyclic code over a finite field F with generator polynomial $g(x)$. So, $\deg g=n-k$, $g(x)$ is monic and divides x^n-1. Let

$$g(x)=g_0+g_1x^1+\cdots+g_{n-k}x^{n-k}+\underbrace{g_{n-k+1}x^{n-k+1}+\cdots+g_{n-1}x^{n-1}}_{0}.$$

We have

① $g_0\neq 0$ since if $g_0=0$, then $x|g(x)$ but this means $x|x^n-1$ which is a contradiction,

② $g_{n-k}=1$ since $g(x)$ is monic,

③ $g_{n-k+1}=0=g_{n-k+2}=g_{n-1}$.

Let $h(x)$ be the polynomial

$$h(x)=\frac{x^n-1}{g(x)}=h_0+h_1x^1+\cdots+h_{k-1}x^{k-1}+h_kx^k+\underbrace{h_{k+1}x^{k+1}+\cdots+h_{n-1}x^{n-1}}_{0}.$$

We have

① $h_k = 1$ since $g(x)$ is monic then so is $(x^n - 1)/g(x) = h(x)$,

② $h_{k+1} = 0 = h_{k+2} = \cdots = h_{n-1}$.

Let $a(x) = a_0 + a_1x^1 + \cdots + a_{n-1}x^{n-1} = gh \mod (x^n - 1)$. By construction, $a(x) = 0$. Equating coefficients of x^i for $i = 1, \ldots, n-1$, we have

$$a_i = 0 = g_0h_i + g_1h_{i-1} + \cdots + g_ih_0 + g_{i+1}h_{n-1} + g_{i+2}h_{n-2} + \cdots + g_{n-1}h_{i-1}.$$

Let

$$g = (g_0, \ldots, g_{n-1}) \text{ and },$$
$$\overline{h} = (h_{n-1}, h_{n-2}, \ldots, h_1, h_0).$$

Then, g is orthogonal to $\overline{h}$ and all the cyclic shifts of $\overline{h}$. So, every cyclic shift of g is orthogonal to every cyclic shift of $\overline{h}$. ◁

Remark 5.3.2: Recall a GM for an (n, k)-cyclic code C is

$$G' = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}_{k\times n} = \begin{bmatrix} \text{——} & g(x) & \text{——} \\ \text{——} & xg(x) & \text{——} \\ & \vdots & \\ \text{——} & x^{k-1}g(x) & \text{——} \end{bmatrix}.$$

Here every row of G' is a right cyclic shift of the row above it. Consider

$$H = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_0 \end{bmatrix}_{(n-k)\times n} = \begin{bmatrix} \text{———} & h^*(x) & \text{———} \\ \text{———} & xh^*(x) & \text{———} \\ & \vdots & \\ \text{——} & x^{n-k-1}h^* & \text{——} \end{bmatrix},$$

where every row of H is a right cyclic shift shift of the row above it as well. We observed that $G'H^\top = 0$.

Let C' be the code spanned by the rows of H. Then, $C' \subseteq C^\perp$. Since $h_k = 1$, then $\operatorname{rank} H = n - k$. So, $\dim C' = n - k = \dim C^\perp$. Hence, $C' = C^\perp$. Hence, H is a PCM for C. ◁

Definition 5.3.3: Let $h(x) = h_0 + h_1x^1 + \cdots + h_kx^k$ be a degree k polynomial. ***Reciprocal polynomial of*** $h(x)$, (simply referred as reciprocal of $h(x)$), denoted by $h_R(x)$, is defined by

$$h_R(x) = h_k + h_{k-1}x^1 + \cdots + h_0x^k = x^k h\left(\frac{1}{x}\right).$$

If $h_0 \neq 0$, then h^* is defined by

$$h^* = h_R \cdot \frac{1}{h_0} = h_R h_0^{-1}.$$

So, h^* is h_R made monic. ◁

Theorem 5.3.4: If C is an (n, k)-cyclic code, then $C^\perp$ is an $(n, n-k)$-cyclic code.

Proof: We have $gh = x^n - 1$. So,

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

Multiplying each side by x^n gives us

$$x^n g\left(\frac{1}{x}\right)h\left(\frac{1}{x}\right) = x^{n-k}g\left(\frac{1}{x}\right)x^k h\left(\frac{1}{x}\right) = g_R(x)h_R(x) = 1 - x^n.$$

Hence, $g_R(x)h_R(x) = -(x^n - 1)$. Hence, $h_R|x^n - 1$ and $g_R|x^n - 1$. So, $h_R(x)$ is a divisor of $x^n - 1$ with degree k. So, the matrix H is a GM for the cyclic code generated by $h^*(x)$. Recall that $h^*(x)$ is $h_R(x)$ made monic. □

5.4 Syndromes in Cyclic Codes

We want to find a more convenient PCM for C. The idea is that we want to

① find a GM for C of the form $[R \mid I_k]_{k\times n}$ (this essentially in standard form),

② so the PCM of C constructed from the GM above is of the form $[I_{n-k} \mid -R^\top]_{(n-k)\times n}$.

Remark 5.4.1: For $i = 0, \ldots, k-1$, long division gives us

$$x^{n-k-i} = \ell_i(x)g(x) + r_i(x),$$

where

- $\deg \ell_i(x) \le k - 1$,
- $\deg g(x) = n - k$,
- $\deg r_i(x) \le n - k - 1$.

Hence, $-r_i(x) + x^{n-k+i} = \ell_i(x)g(x) \in C$. Let

$$G = \left[\begin{array}{c} \text{——} - r_0(x) + x^{n-k} \text{——} \\ \text{——} - r_1(x) + x^{n-k+1} \text{——} \\ \vdots \\ \text{——} - r_{k-1}(x) + x^{n-1} \text{——} \end{array}\right] = \left[\begin{array}{c|cccc} \text{——} - r_0(x) \text{——} & 1 & 0 & \cdots & 0 \\ \text{——} - r_1(x) \text{——} & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \text{——} - r_{k-1}(x) \text{——} & 0 & \cdots & 0 & 1 \end{array}\right] = [R \mid I_k].$$

◁

G has rank k, so G is a GM for C. Then, by Theorem 3.2.9, $H = [I_{n-k} \mid -R^\top]_{(n-k)\times n}$ is a PCM for H since $C = (C^\perp)^\perp$. We have

$$H^\top = \left[\frac{I^{n-k}}{-R}\right] = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ \hline & & & \\ & r_0(x) & & \\ & r_1(x) & & \\ & \vdots & & \\ & r_{k-1}(x) & & \end{array}\right] = \left[\begin{array}{c} x^0 \mod g(x) \\ x^1 \mod g(x) \\ \vdots \\ x^{n-k-1} \mod g(x) \\ \hline x^{n-k} \mod g(x) \\ x^{n-k-1} \mod g(x) \\ \vdots \\ x^{n-1} \mod g(x) \end{array}\right].$$

Hence, any i^{th} column of H (i^{th} row of $H^\top$) is $x^i \mod g(x)$ for $i = 0, \ldots, n-1$. Hence, if $r = (r_0, \ldots, r_{n-1}) \in V_n(F)$, then

$$\begin{aligned} s = Hr^\top &= (r_0 x^0 \mod g(x)) + (r_1 x^1 \mod g(x)) + \cdots + (r_{n-1} x^{n-1} \mod g(x)) \\ &= (r_0 x^0 + r_1 x^1 + \cdots + r_{n-1} x^{n-1}) \mod g(x) \\ &= r(x) \mod g(x). \end{aligned}$$

Theorem 5.4.2: Let C be a cyclic code with generator polynomial $g(x)$ and let $r \in V_n(F)$. Then, the syndrome of r (with respect to the previous PCM) is $s(x) = r(x) \mod g(x)$.

Proof: This is explicitly shown above. □

Example 5.4.3: It is easy to see that $g(x) = 1 + x + x^2 + x^3 + x^6$ is a generator polynomial for a $(15, 9)$-binary cyclic code since $\deg g = 15 - 9 = 6$ and since $g(x) | x^{15} - 1$. To compute the syndrome of

$$r = (11101\ 11011\ 00000),$$

we write r in polynomial form and use long division to find $\ell(x)$ and $\text{rem}(x)$ (quotient and remainder polynomials) such that

$$r(x) = \ell(x)g(x) + \text{rem}(x).$$

We have $r(x) = x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1$ and since

$$x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1 = (x^6 + x^3 + x^2 + x + 1)(x^3 + x^2) + (x^5 + x^4 + x + 1),$$

we find $\ell(x) = x^3 + x^2$ and $\text{rem}(x)$ as $x^5 + x^4 + x + 1$. So, $s(x) = 1 + x + x^4 + x^5$. In vector form we have $s = (110011)$. ◁

Remark 5.4.4: This method of finding the syndrome, that is, finding $s(x) = \text{rem}(x) \mod g(x)$ can be implemented in hardware using very a very efficient and fast circuit. ◁

Remark 5.4.5: Given the syndrome s of r, the syndromes of cyclic shifts of r can be easily computed. ◁

Theorem 5.4.6: Let $r \in V_n(F)$ and $s(x) = r(x) \mod g(x) = s_0 + s_1 x^1 + \cdots + s_{n-k-1} x^{n-k-1}$. The syndrome of $xr(x)$ (the right cyclic shift of $r(x)$) is

$$xs(x) - s_{n-k-1} g(x).$$

Proof: By definition we have $r(x) = \ell(x)g(x) + s(x)$. Hence, $xr(x) = x\ell(x)g(x) + xs(x)$. We consider two cases.

- Case 1: $\deg s < n-k-1$. So, $s_{n-k-1} = 0$. Then, $\deg(x(s)) < n-k$. Hence we have

$$xr(x) \mod g(x) = xs(x).$$

  Hence, syndrome of r is

$$xs(x) - s_{n-k-1}g(x) = xs(x).$$

- Case 2: $\deg s = n-k-1$. So, $s_{n-k-1} \neq 0$. Then,

$$\begin{aligned} xr(x) &= x\ell(x)g(x) + xs(x) + s_{n-k-1}g(x) - s_{n-k-1}g(x) \\ &= g(x)(x\ell(x) + s_{n-k-1}) + \underbrace{xs(x) - s_{n-k-1}g(x)}_{\text{has degree at most } n-k-1}. \end{aligned}$$

  Hence, $xr(x) \mod g(x) = xs(x) - s_{n-k-1}g(x)$. Hence, syndrome of $r(x)$ is $xs(x) - s_{n-k-1}g(x)$.

□

5.4.1 Burst Error Correcting

We will see that cyclic codes are good for cyclic burst error correcting.

Definition 5.4.7: Let $e \in V_n(F)$. The ***cyclic burst length of*** e is the length of the smallest cyclic block that contains all the non-zero entries of e. We say e has *cyclic burst error length* t if its cyclic burst length is t. ◁

Example 5.4.8: Consider

$$V_9(\mathbb{Z}_2) \ni e = (011000001) = (e_1e_2e_3e_4e_5e_6e_7e_8e_9).$$

So, the non-zero entries of e are e_2, e_3 and e_9 but the smallest cyclic block that contains the non-zero entries of e is $e_9e_1e_2e_3e_4$. Hence, cyclic burst length of e is 4. ◁

Definition 5.4.9: A linear code C is called a ***t-cyclic burst error correcting code*** if every cyclic burst error of length at most t lies in a unique coset of C. Longest such t is called the ***cyclic burst error capability of*** C. ◁

Example 5.4.10: $g(x) = 1 + x + x^2 + x^3 + x^6$ generates a $(15, 9)$-binary cyclic code C. C is a 3-cyclic burst error correcting code. We have $d(C) \leq 5$ (since its generator polynomial has 5 terms), so $e \leq 2$. To verify this we can check that each cyclic burst error of length at most 3 has a unique syndrome.

This example was explicitly solved in section 5.7 as example 20 in textbook, on pages 176–178. ◁

Example 5.4.11: $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ generates a $(15, 7)$-binary cyclic code that is 4-cyclic burst error correcting. We know it has distance at most 5, so $e \leq 2$. ◁

Remark 5.4.12: To construct codes with high cyclic burst error correcting capability, we can

① use computer search,

② use RS codes,

③ use codeword interleaving. ◁

Theorem 5.4.13: Let C be an (n, k, d)-code (linear) over GF(q). Let t be its cyclic burst error correcting capability. Then,

$$\left\lfloor \frac{d-1}{2} \right\rfloor \leq t \leq n - k.$$

Proof: We know that every cyclic burst of length at most t has weight (number of non-zero indices) at most t. Since $d(C) = w(C)$ and since any codeword in C of weight at most $\lfloor \frac{d-1}{2} \rfloor$ has a unique syndrome, then $\lfloor \frac{d-1}{2} \rfloor \leq t$. Moreover, the number of cyclic burst errors where the non-zero entries lie in the first t coordinate positions is q^t. Each of them has a unique coset and the total number of cosets is q^{n-k}. So, $q^t \leq q^{n-k}$. Hence, $t \leq n - k$. □

Exercise 5.4.14: Show that $t \leq \frac{n-k}{2}$. This is known as the Rieger bound. ◁

5.4.1.1 Error Trapping Decoding for Cyclic Burst Errors

Remark 5.4.15: Recall that if C is an (n, k)-cyclic code over F with generator polynomial $g(x)$ and if C is a t-cyclic burst error correcting code then $t \leq n - k$. Also, $H = [I_{n-k} \mid -R^\top]$ is a PCM for C and the syndrome of $r \in V_n(F)$ is $s(x) = r(x) \mod g(x)$.
Let e be a cyclic burst of length at most t. To decode cyclic burst errors, such as e, we will compute shifts of e, say $e_i = x^i e$. ◁

Let $r \in V_n(F)$ be the received vector with polynomial representation $r(x)$. Let $s_i(x)$ be syndrome of $x^i r(x)$ for $i = 0, \ldots, n-1$. So we have

$$s_i(x) = x^i r(x) \mod g(x)$$

and $s_0 = r(x) \mod g(x)$. We use the following algorithm for error trapping

Algorithm 5.4.16: Error Trapping

```
1 for i = 0, ..., n − 1 do
2     Compute s_i(x) as described above
3     if s_i(x) is a (non-cyclic) burst of length at most t then
4         e_i(x) ← (s_i(x), 0) (this is s_i(x) concatenated with zeros on the right)
5         e(x) ← x^{n−i} e_i(x)
6         Decode r(x) to r(x) − e(x)
7 reject r(x)
```

Note that in line 2, if $i = 0$, then we use polynomial division, otherwise s_i is s_{i-1} shifted.

Example 5.4.17: Recall $g(x) = 1 + x + x^2 + x^3 + x^6$ is a generator polynomial for a $(15, 9)$-binary cyclic code with cyclic burst error correcting capability 3 (as solved in pages 176–178 in textbook). To decode

$$r = (11101\ 11011\ 00000),$$

we find $s_i(x) = x^i r(x) \mod g(x)$. We have

i	$s_i(x)$
0	110011
1	100101
2	101110
3	010111
4	110111
5	100111
6	101111
7	101011
8	101011
9	101000

We stop at $s_9(x)$ since $s_9(x)$ is a non-cyclic burst of at most 3. So,

$$e = x^{15-9} e_9 = x^6 e_9 = (00000\ 01010\ 00000).$$

Hence, we correct r to c where

$$c = r - e = (11101\ 10001\ 00000). \qquad \triangleleft$$

5.4.1.2 Interleaving Codewords

Our goal is to improve the cyclic burst error correcting capability (c.b.e.c.c.) of a code C. Suppose C is an (n, k)-code with c.b.e.c.c. t. Suppose the following codewords $v_1, \ldots, v_s$ are transmitted.

$$\begin{aligned} v_1 &= (v_{11}, v_{12}, \ldots, v_{1n}), \\ v_1 &= (v_{21}, v_{22}, \ldots, v_{2n}), \\ &\vdots \qquad\qquad \vdots \\ v_s &= (v_{s1}, v_{s2}, \ldots, v_{sn}). \end{aligned}$$

Suppose $v_1, \ldots, v_s$ are transmitted in this order. So the data in v_{11} is received first, and then v_{12} until v_{1n}. Then, v_{21}, v_{22} etc. until v_{sn}. We can represent it in a big vector as follows.

$$\big[v_{11}, v_{12}, \ldots, v_{1n}, v_{21}, v_{22}, \ldots, v_{2n}, \ldots, v_{s1}, v_{s2}, \ldots, v_{sn}\big].$$

If a cyclic burst of length at most t occurs in any codeword, that error can be corrected. Instead, we transmit the data with the order of the columns and represent it in a big (fat) codeword as follows.

$$\big[v_{11}, v_{21}, \ldots, v_{s1}, v_{12}, v_{22}, \ldots, v_{s2}, \ldots, v_{1n}, v_{2n}, \ldots, v_{sn}\big].$$

Hence, the first and the last data transmitted are the same in both orders. Now, if a cyclic burst error of length at most $s \cdot t$ occurs in this fat codeword, this means each original codeword suffered a cyclic burst error of length at most t.

Example 5.4.18: Recall as covered in Example 5.4.10 and Example 5.4.17, $g(x) = 1 + x + x^2 + x^3 + x^6$ is the generator polynomial for a $(15, 9)$-binary cyclic code with 3-c.b.e.c.c. We interleave C to depth $s = 100$ and obtain the code C^*. C^* is a $(1500, 900)$-binary cyclic code with 300-c.b.e.c.c. with generator polynomial

$$g(x^{100}) = 1 + x^{100} + x^{200} + x^{300} + x^{600}. \qquad \triangleleft$$

5.5 BCH Codes and Minimal Polynomials

Recall 5.5.1: We can view the field $F = \mathrm{GF}(p^m)$ as a vector space of dimension m over $\mathbb{Z}_p$ where $\mathbb{Z}_p$ is a subfield of F. More generally, for any prime power q, we can view the finite field $\mathrm{GF}(q^m)$ as a vector space of dimension m over $\mathrm{GF}(q)$ and $\mathrm{GF}(q)$ as a subfield of $\mathrm{GF}(q^m)$. ◁

Example 5.5.2: We can view $\mathrm{GF}(2^{16})$ as follows.

$\mathrm{GF}(2^{16})$ is a vector space of dimension 16 over $\mathrm{GF}(2)$,
$\mathrm{GF}(2^{16})$ is a vector space of dimension 8 over $\mathrm{GF}(2^2)$,
$\mathrm{GF}(2^{16})$ is a vector space of dimension 4 over $\mathrm{GF}(2^4)$,
$\mathrm{GF}(2^{16})$ is a vector space of dimension 2 over $\mathrm{GF}(2^8)$, and
$\mathrm{GF}(2^{16})$ is a vector space of dimension 1 over $\mathrm{GF}(2^{16})$. ◁

Remark 5.5.3: We call $\mathrm{GF}(q^m)$ the *extension field* and $\mathrm{GF}(q)$ as the *subfield*. Informally, $\mathrm{GF}(q^m)$ is the big field and $\mathrm{GF}(q)$ is the small field contained within big field.

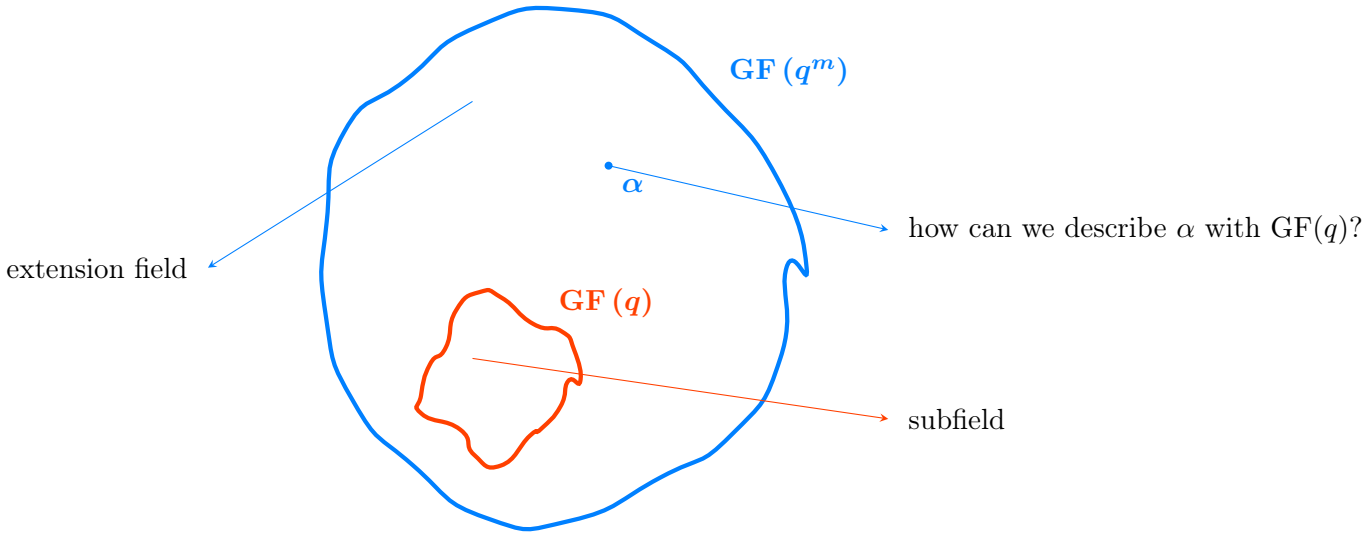

Figure 5.5.1: Extension field, $\mathrm{GF}(q^m)$ and subfield, $\mathrm{GF}(q)$ with $\alpha \in \mathrm{GF}(q^m)$.

◁

5.5.1 Minimal Polynomials

Definition 5.5.4: Let $\alpha \in \mathrm{GF}(q^m)$. The ***minimal polynomial of*** α ***over*** $\mathrm{GF}(q)$, denoted $m_\alpha(x)$, is the (so it's unique) monic polynomial of smallest degree in $\mathrm{GF}(q)[x]$ such that $m_\alpha(\alpha) = 0$. ◁

Example 5.5.5: For $0 \in \mathrm{GF}(q^m)$ we have $m_0(x) = x$. ◁

Example 5.5.6: Consider $\mathrm{GF}(2^2) = \mathbb{Z}_2[x]/(x^2 + x + 1) = \{0, 1, x, x + 1\}$. We can find minimal

polynomials of every $\alpha \in \mathrm{GF}(2^2)$ over $\mathrm{GF}(2)$ as follows.

$$\begin{aligned} m_0(y) &= y, \\ m_1(y) &= y - 1, \\ m_x(y) &= y^2 + y + 1, \\ m_{x+1}(y) &= y^2 + y + 1. \end{aligned}$$ ◁

5.5.1.1 Properties of Minimal Polynomials

Remark 5.5.7: We know minimal polynomial for $0 \in \mathrm{GF}(q^m)$ exists. It's also clear that for any non-zero $\alpha \in \mathrm{GF}(q^m)$, minimal polynomial of α, $m_\alpha(x)$, exists. Let $\alpha \in \mathrm{GF}(q^m)$ be non-zero and let $\mathrm{ord}(\alpha) = t$. So $t|(q^m - 1)$ and $\alpha^t = 1$. Then α is a root of $x^t - 1 \in \mathrm{GF}(q)[x]$. Moreover, if $f(x) \in \mathrm{GF}(q)[x]$ with $f(\alpha) = 0$, then if $c \in \mathrm{GF}(q)$ is the leading coefficient of f, then $f'(x) = c^{-1}f(x) \in \mathrm{GF}(q)[x]$, $f'(x)$ is monic and $f'(\alpha) = 0$. Hence, $m_\alpha(x)$ exists. ◁

Theorem 5.5.8 (Properties of minimal polynomial)**:** Let $\alpha \in \mathrm{GF}(q^m)$ with minimal polynomial $m_\alpha(x)$. Then,

1. $m_\alpha(x)$ is unique,
2. $m_\alpha(x)$ is irreducible over $\mathrm{GF}(q)$,
3. $\deg(m_\alpha(x)) \leq m$,
4. if $f(x) \in \mathrm{GF}(q)[x]$, then $f(\alpha) = 0$ if and only if $m_\alpha(x)|f(x)$.

Proof:

1. Suppose, for contradiction, there exists distinct $m_1(x), m_2 \in \mathrm{GF}(q)[x]$ such that both m_1 and m_2 are minimal polynomials for some $\alpha \in \mathrm{GF}(q^m)$. So both $m_1(x)$ and $m_2(x)$ are monic and they have the same smallest degree. Consider $r(x) = m_1(x) - m_2(x)$. Then

   $$r(\alpha) = m_1(\alpha) - m_2(\alpha) = 0 - 0 = 0.$$

   But this means $r(x)$ is a non-zero polynomial with $\deg r(x) < \deg m_1$ which is a contradiction. Hence, $m_1(x) = m_2(x)$.

2. Suppose, for contradiction, $m_\alpha(x)$ is reducible over $\mathrm{GF}(q)$. Then we can write

   $$m_\alpha(x) = s(x)t(x),$$

   for some $s(x), t(x) \in \mathrm{GF}(q)[x]$ with $\deg s(x), \deg t(x) \leq \deg m_\alpha(x)$. Then

   $$m_\alpha(\alpha) = 0 = s(\alpha)t(\alpha).$$

   Hence, either $s(\alpha) = 0$ or $t(\alpha) = 0$ which contradicts the minimality of $\deg m_\alpha(x)$. Hence, $m_\alpha(x)$ is irreducible over $\mathrm{GF}(q)$.

3. Recall $\mathrm{GF}(q^m)$ is a vector space of dimension m over $\mathrm{GF}(q)$. Hence, the $m + 1$ elements

   $$1, \alpha, \ldots, \alpha^m$$

are linearly dependent over GF(q). So we can write

$$a_0 + a_1\alpha + \cdots + a_m\alpha^m = 0,$$

for some $a_0, a_1, \ldots, a_m \in \mathrm{GF}(q)$ where $a_0, \ldots, a_m$ are not all zero. Hence, α is a root of the non-zero polynomial

$$a_0 + a_1\alpha + \cdots + a_m\alpha^m \in \mathrm{GF}(q)[x].$$

This polynomial have at most degree m. Hence, minimal polynomial has at most degree m.

④ Let $f(x) \in \mathrm{GF}(q)[x]$. Using the division algorithm for polynomials, we can write

$$f(x) = \ell(x)m_\alpha(x) + r(x),$$

where $\ell(x), r(x) \in \mathrm{GF}(q)[x]$ and $\deg r(x) < \deg m_\alpha(x)$. We have

$$f(\alpha) = \ell(\alpha)m_\alpha(\alpha) + r(\alpha) = r(\alpha).$$

Hence,

$$\begin{aligned} f(\alpha) = 0 &\iff r(\alpha) = 0 \\ &\iff r(x) = 0(\text{ since } \deg r(x) < \deg m_\alpha(x)) \\ &\iff m_\alpha(x) | f(x). \end{aligned}$$

□

5.5.1.2 Formula for Calculating Minimal Polynomials

We will derive a formula for computing $m_\alpha(x)$ of $\alpha \in \mathrm{GF}(q^m)$ over GF(q). We will see that roots of $m_\alpha(x)$ are precisely the conjugates of α with respect to GF(q).

Theorem 5.5.9: Let $\alpha \in \mathrm{GF}(q^m)$. Then $\alpha \in \mathrm{GF}(q)$ if and only if $\alpha^q = \alpha$.

Proof: Since $\alpha^q = \alpha$ for all $\alpha \in \mathrm{GF}(q)$, the elements of GF($q$) are roots of the polynomial $x^q - x$. Since $\deg x^q - x = q$, it cannot have any other roots in GF(q^m). Hence, $\alpha \in \mathrm{GF}(q)$ if and only if $\alpha^q = \alpha$. □

Definition 5.5.10: Let $\alpha \in \mathrm{GF}(q^m)$. Let t be the smallest positive integer such that $\alpha^{q^t} = \alpha$ (note that $t \leq m$). Then ***the set of conjugates of*** α ***with respect to*** GF(q) is

$$C(\alpha) = \{\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{t-1}}\}.$$

Note that the elements of $C(\alpha)$ are distinct. ◁

Theorem 5.5.11: Let $\alpha \in \mathrm{GF}(q^m)$. Then the minimal polynomial of α over GF(q) is

$$m_\alpha(x) = \prod_{\beta \in C(\alpha)} (x - \beta) = (x - \alpha)(x - \alpha^q)(x - \alpha^{q^2}) \cdots (x - \alpha^{q^{t-1}}).$$

Proof: We have that $m_\alpha(x)$ is monic and $m_\alpha(\alpha) = 0$. Let $m_\alpha(x) = \sum_{i=0}^{t} m_i x_i$. The coefficients m_i are in $\mathrm{GF}(q^m)$. We need to prove that $m_i \in \mathrm{GF}(q)$. We have

$$\begin{aligned} m_\alpha(x)^q &= \prod_{\beta \in C(\alpha)} (x-\beta)^q \\ &= \prod_{\beta \in C(\alpha)} (x^q - \beta^q) \\ &= \prod_{\beta \in C(\alpha)} (x^q - \beta), \text{ since } C(\alpha) = \{\beta^q \mid \beta \in C(\alpha)\} \\ &= m_\alpha(x^q) \\ &= \sum_{i=0}^{t} m_i x^{iq}. \end{aligned} \tag{5.5.1}$$

Moreover,

$$m_\alpha(x)^q = \left(\sum_{i=0}^{t} m_i x^i \right)^q = \sum_{i=0}^{t} m_i^q x^{iq}. \tag{5.5.2}$$

Comparing coefficients of x^{iq} in (5.5.1) and (5.5.2) gives $m_i = m_i^q$ for all $0 \leq i \leq t$. Hence, $m_i \in \mathrm{GF}(q)$ and so $m_\alpha(x) \in \mathrm{GF}(q)[x]$.

Now, let $f(x) \in \mathrm{GF}(q)[x]$ be non-zero and suppose that $f(\alpha) = 0$. Let $f(x) = \sum_{i=0}^{d} f_i x^i$. Then,

$$f(\alpha^q) = \sum_{i=0}^{d} f_i \alpha^{iq} = \left(\sum_{i=0}^{d} f_i \alpha^i \right)^q = f(\alpha)^q = 0.$$

Hence, the elements of $C(\alpha)$ are roots of $f(x)$. Since the roots of $m(x)$ are precisely the elements of $C(\alpha)$, we conclude that $m_\alpha(x)$ is the monic polynomial of smallest degree in $\mathrm{GF}(q)[x]$ that has α as a root. □

Example 5.5.12: Consider $\mathrm{GF}(2^4) = \mathbb{Z}_2[x]/(x^4 + x + 1)$. Find the minimal polynomial of $\beta = x^2 + x^3$ over $\mathbb{Z}_2$. So we have $q = 2$ and $m = 4$.

It would help to have a generator α of $\mathrm{GF}(2^4)^*$ and its powers. It turns out that $\alpha = x$ is a generator, as the following table shows.

$\alpha^0 = 1$	$\alpha^5 = \alpha + \alpha^2$	$\alpha^{10} = 1 + \alpha + \alpha^2$
$\alpha^1 = \alpha$	$\alpha^6 = \alpha^2 + \alpha^3$	$\alpha^{11} = \alpha + \alpha^2 + \alpha^3$
$\alpha^2 = \alpha^2$	$\alpha^7 = 1 + \alpha + \alpha^3$	$\alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3$
$\alpha^3 = \alpha^3$	$\alpha^8 = 1 + \alpha^2$	$\alpha^{13} = 1 + \alpha^2 + \alpha^3$
$\alpha^4 = 1 + \alpha$	$\alpha^9 = \alpha + \alpha^3$	$\alpha^{14} = 1 + \alpha^3$

We have $\beta = x^2 + x^3 = \alpha^6$. Hence, $C(\beta) = \{\alpha^6, \alpha^{12}, \alpha^9, \alpha^3\}$. Note that $\alpha^{24} = \alpha^9$ since the order of

α is 15. Hence,

$$\begin{aligned} m_\beta(y) &= (y-\alpha^3)(y-\alpha^6)(y-\alpha^9)(y-\alpha^{12}) \\ &= [y^2+(\alpha^3+\alpha^6)y+\alpha^9][y^2+(\alpha^9+\alpha^{12})y+\alpha^6] \\ &= [y^2+\alpha^2 y+\alpha^9][y^2+\alpha^8 y+\alpha^6] \\ &= y^4+(\alpha^2+\alpha^8)y^3+(\alpha^9+\alpha^{10}+\alpha^6)y^2+(\alpha^8+\alpha^2)y+1 \\ &= y^4+y^3+y^2+y+1. \end{aligned}$$

We see that the coefficients of $m_\beta(y)$ are in GF(2). Also we simplified terms such as $\alpha^3+\alpha^6$ to α^2 by using the table of powers of α. ◁

Due COVID-19, lectures 31-36 are held as video lectures. Instructor posted 10 lectures on LEARN (V0-V9) and posted the following information on Piazza at https://piazza.com/class/k4ht3u7bp46bf?cid=43:

Video lectures (updated Apr 3)

- March 22: I have posted three video lectures V0, V1, V2 on "Factoring x^n-1 over GF(q)" on LEARN. The intent is that you view these lectures between Mar 23-25.
- March 24: Video lectures V3, V4 on "BCH codes". Please view these lectures between Mar 25-29.
- March 25: Video lectures V5, V6 on "BCH decoding". Please view these lectures between Mar 30-Apr 1.
- March 26: Video lecture V7 on "Reed-Solomon codes". Please view this lecture between Apr 1-3.
- April 2: Video lecture V8: This will be a short wrap-up. Please view this lecture between Apr 2-3.
- April 3: Video lecture V9 on "code-based public-key encryption" (optional viewing).

These video lectures are not very polished – It's clear that I am not yet ready to become a YouTuber.[1] I have had to learn a lot of new technologies in the past week. If you have any suggestions on how I could improve the quality of the video lectures, please do email them to me.

Questions? Please use the V# threads on Piazza to ask questions about the video lectures. If appropriate, you can include in your question the relevant time stamp from the lecture. You can also ask questions about the lecture in office hours.

#pin

The content covered on these video lectures will not be typeset.

This concludes the final lecture(s) for CO 331 in Winter 2020.

[1] This is wrong. Video lectures were actually pretty good, a lot better than many lecture videos on YouTube.

Index

A
alphabet, 3

B
binary operation, 10
binary symmetric channel (BSC), 4

C
code, 3
  block, 3
    rate of, 5
  cyclic, 35
  equivalent, 21
  linear, 19
  perfect, 24
  self dual, 30
  self orthogonal, 30
  systematic, 20
codeword, 3
congruence in polynomials, 14
congruency of codewords, 27
coset leader, 28
coset of $V_n(F)$, 27
cyclic burst (error) length of a vector, 44
cyclic burst error capability of a code, 44
cyclic subspace, 35

D
dot product, 21
dual code, 22

E
e-error correcting code, 7
e-error detecting code, 7
error vector, 26

F
field, 10
  characteristic of, 11
  Galois field, 16
  isomorphic, 12
  order
    of element, 17
  set of polynomials over a field, 14
  subfield, 12

G
generator, 18
generator matrix, 20
  standard, 20

H
Hamming code, 25
Hamming distance, 5
Hamming weight, 19

I
ideal, 35
  generated by an element, 36
  principal, 36
inner product, 21
irreducible, 15

M
minimal polynomial, 47

O
order, 10
  additive order of an element, 10
  multiplicative order of an element, 10
  of a field, 10
orthogonal vectors, 21

P
parity-check matrix, 23
primitive element, 18

R
reciprocal polynomial, 41
replication code, 1
ring, 10
  commutative, 10
  division ring, 10
  principal ideal ring, 36
  with identity, 10

S
sphere packing problem, 8, 24
symbol error probability, 4
syndrome, 26

T
t-cyclic burst error correcting code, 44
the generator polynomial of I, 36
the set of conjugates, 49

W
word, 3
    length of, 3