CIT413045 Summer 25

# Coding Theory

*Lecturer: Prof. Dr. Violetta Weger*

These lecture notes will be consistently updated before the lectures. If you find any typos, please send them to me via email.

**Overview** Coding Theory emerged in the 1940's with the fundamental paper "A mathematical theory of communication" by Shannon [17].

Shannon was studying therein the properties of communication channels prone to errors. While Shannon's focus lays on the channel and its capacity, Hamming [7] started focusing on the algebraic properties of the code used to correct errors. Thus, the field has quickly split into two main parts: Information theory and algebraic coding theory.

This lecture is complementary to the lecture "Channel Coding", as we will solely focus, just like Hamming, on the algebraic properties of codes.

Algebraic coding theory is a very young mathematical subject and is full of open problems and possible research questions. Due to the error-correcting capabilities of codes, they find numerous applications in the real world, such as: reliable communication, data storage, distributed and coded computing, information retrieval, network coding, cryptography and some of the newest ones include DNA storage and quantum computing.

You will learn about the parameters, properties and relations between codes, the main constructions of codes, some decoding algorithms and see different applications.

Disclaimer: This course is **not** about programming.

## Administrative Information

The lectures will be held

- Tuesdays, 16:15-17:45 in MI 00.07.014
- Wednesday, 14:15-15:45 in MI 00.07.014

The tutorials will be held

- Mondays, 10:15-11:45 in MI 03.08.011

**Exercises:** I will upload exercise sheets on Moodle, which are voluntary to solve, but upon 70% of correct completion, I will provide a grade bonus of $+0.3$.

**Exam:** The exam will either be in written (90 min) or oral (30 min) form, depending on the number of students.

**Prerequisites:**

- MA0004 Linear Algebra 1,
- MA0005 Linear Algebra 2 and Discrete Structures.

**Material:** Most of the content of these lecture notes is based on

- the book *Fundamentals of Error-Correcting Codes* by W.C. Hufmann and V. Pless [8],
- the book *The Theory of Error-Correcting Codes* by N.J.A Sloane F.J. MacWilliams [10],
- the book *Introduction to Coding Theory* by J. van Lint [19] and
- the book *Introduction to Coding Theory* by R. Roth [15].

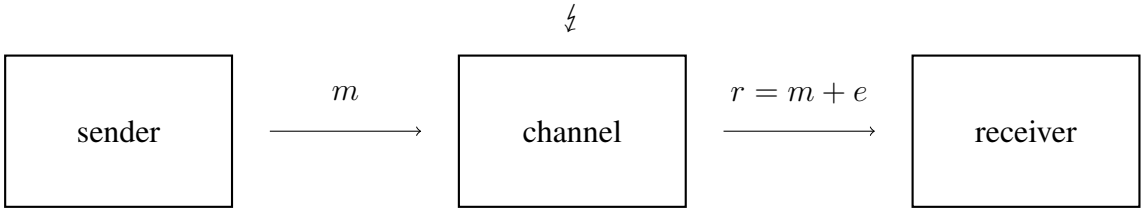
**Schedule**

In green we marked the holidays and in red the days I am not here. On these days we will either do a "flipped classroom", where you will be provided with video lectures to watch and we can discuss questions at the next in-person event or there will be a substitute lecturer.

| Tutorial Monday | Hand in Sheet | New Sheet to Discuss | Lecture Tuesday | Lecture Wednesday |
|---|---|---|---|---|
| 21.04 | | | 22.04 | 23.04 |
| 28.04 | | Sheet 1 | 29.04 | 30.04 |
| 05.05 | Sheet 1 | Sheet 2 | 06.05 | 07.05 |
| 12.05 | Sheet 2 | Sheet 3 | 13.05 | 14.05 |
| 19.05 | Sheet 3 | Sheet 4 | 20.05 | 21.05 |
| 26.05 | Sheet 4 | Sheet 5 | 27.05 | 28.05 |
| 02.06 | Sheet 5 | Sheet 6 | 03.06 | 04.06 |
| 09.06 | | | 10.06 | 11.06 |
| 16.06 | Sheet 6 | Sheet 7 | 17.06 | 18.06 |
| 23.06 | Sheet 7 | Sheet 8 | 24.06 | 25.06 |
| 30.06 | Sheet 8 | Sheet 9 | 01.07 | 02.07 |
| 07.07 | Sheet 9 | Sheet 10 | 08.07 | 09.07 |
| 14.07 | Sheet 10 | Summary | 15.07 | 16.07 |
| 21.07 | | Summary | 22.07 | 23.07 |

Let us start with some motivation for this course.

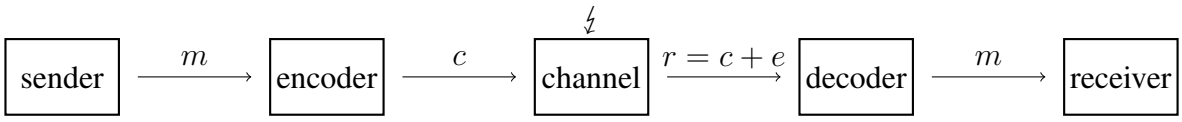
Whenever we communicate digitally, we do so over *noisy* channels. This means that the channel will introduce some errors, e.g. flip some bits of the sent message.




Assume the *sender* sends a message in binary $m = (1, 0, 1)$ over a noisy channel and the *receiver* receives $(0, 0, 1)$. The most natural way is to interpret this received vector as $m + e$, where $e = (1, 0, 0)$ is called the *error vector*.

The aim of coding theory is to enable reliable communication. As we cannot alter the channel we have at hand, we alter what we send. Instead of sending the message directly, we first *encode* it. In theory, you could think of this process as adding redundancy to your message.

In our toy example, instead of sending $m = (1, 0, 1)$ we could send $c = (1, 0, 1, 1, 0, 1, 1, 0, 1)$. The channel might introduce an error, however, our encoding enforces possible sent vectors (called *codewords*) to be distant, so that upon receiving $r = (1, 0, 0, 1, 0, 1, 1, 0, 1)$, we can find only one closest codeword, $c$ and thus in turn $m$.




The code should also come with an efficient *decoder*, that is an algorithm which upon receiving $r$ outputs the closest codeword and knowing the encoding function also the sent message. However, there are many rules to this game: We cannot correct any amount of errors, which leads us to consider several bounds on the code parameters and in order to get efficient algorithms we need to construct codes with algebraic structure.

Let us have a quick glance at the highlights of this lecture.

**Theorem 0.1** (Optimality of Random Codes)**.** *Random codes attain with high probability, for $n$ going to infinity, the Gilbert-Varshamov bound, and for $q$ going to infinity, the Singleton bound.*

Interestingly, the most famous conjecture in coding theory is older than coding theory itself (due to the connection to finite geometry) [16].

**Open Question 0.2** (MDS Conjecture)**.** *Let $q$ be odd, then there is no code attaining the Singleton bound with $n > q + 1$.*

This has recently been proven for $q$ being prime, by Ball [2].

**Theorem 0.3** (Perfect Codes)**.** *The only known perfect codes are the Hamming codes and the two Golay codes.*

**Theorem 0.4** (Constant-Weight Codes)**.** *Every constant-weight code is an $\ell$-fold duplication of the Simplex code.*

**Theorem 0.5** (MacWilliams Identities)**.** *The weight distribution of a code is completely determined by that of the dual code.*

**Theorem 0.6** (SDP is NP-hard)**.** *To decode a random code is one of the hardest problems in mathematics.*

Clearly, these statements need to be refined and the bounds and codes introduced, as we plan to do in this lecture.

# Contents


**1 Finite Fields** **10**
- 1.1 Summary . . . 10
- 1.2 Prime Fields . . . 11
- 1.3 Subfields and Field Extensions . . . 12
- 1.4 Construction of Finite Fields . . . 13
- 1.5 Multiplicative Group . . . 16
- 1.6 Uniqueness . . . 17
- 1.7 Different Representation . . . 18
- 1.8 Trace and Norm . . . 22
- 1.9 Some Properties . . . 24
- 1.10 Invertible Matrices . . . 25

**2 Basics of Codes** **27**
- 2.1 Generator Matrix and Parity-Check Matrix . . . 27
- 2.2 Hamming Metric . . . 35
- 2.3 Error-Correction Capability . . . 37
- 2.4 Subcodes and Supercodes . . . 40
- 2.5 Counting Codes . . . 42

**3 MDS Codes** **44**
- 3.1 Singleton Bound . . . 44
- 3.2 Trivial MDS Codes . . . 45
- 3.3 Reed-Solomon Codes . . . 46
- 3.4 Generalized Reed-Solomon Codes . . . 48
- 3.5 Primitive Reed-Solomon Codes . . . 51
- 3.6 MDS Conjecture . . . 53
- 3.7 Decoding of RS Codes . . . 57
  - 3.7.1 Berlekamp-Welch . . . 57

**4 Sphere-Packing and Sphere-Covering** **61**
- 4.1 Hamming bound . . . 61
- 4.2 Perfect Codes . . . 62
- 4.3 Asymptotic Hamming bound . . . 65
- 4.4 Gilbert-Varshamov Bound . . . 67
- 4.5 Asymptotic Gilbert-Varshamov Bound . . . 68

**5 Plotkin Bound** **71**
- 5.1 Simplex Code . . . 75
- 5.2 Asymptotic Bound . . . 77
- 5.3 Comparison of the Bounds . . . 78
- 5.4 Quick overview of other bounds . . . 78

**6 Construction of New Codes** **80**
6.1 Extension of Codes . . . 80
6.2 Puncturing . . . 82
6.3 Shortening . . . 83
6.4 Product of codes . . . 85
6.5 Plotkin Sum . . . 86
6.6 Sum of Codes . . . 88
6.7 Intersection of Codes . . . 88
6.8 Expansion Codes . . . 90
6.9 Subfield Subcodes . . . 94
6.9.1 Alternant Codes . . . 96
6.9.2 Goppa Codes . . . 97
6.10 Trace Codes . . . 99
6.11 Power Codes . . . 102
6.12 Concatenation . . . 104

**7 Equivalence of Codes** **106**
7.1 Invariants . . . 110
7.2 Closure . . . 113

**8 Cyclic Codes** **116**
8.1 Polynomial Representation . . . 117
8.2 Duality . . . 120
8.3 Cyclotomic Classes . . . 123
8.4 Generalizations . . . 125

**9 Generic Decoding** **128**
9.1 Interlude: Code-based Cryptography . . . 128
9.2 Decoding Problem . . . 129
9.3 Solvers . . . 130
9.4 ISD Algorithms . . . 133
9.4.1 Prange's Algorithm . . . 135
9.5 Stern's Algorithm . . . 137
9.5.1 Asymptotic Cost . . . 140

**10 List Decoding** **145**
10.1 Johnson Bound . . . 147
10.2 Bivariate Polynomials . . . 150
10.3 Recap on Berlekamp-Welch . . . 151
10.4 Sudan's Algorithm . . . 151

**11 MacWilliams Identity** **154**
11.1 Characters . . . . . 154
11.2 Schur Orthogonality . . . . . 155
11.3 Krawtchouk Coefficients . . . . . 156
11.4 Linear Programming Bound . . . . . 159

**12 Rank-Metric Codes** **161**

# Notation

Since we cannot include everything, we will assume a certain background. For example, we assume that modular arithmetic, the notion of field, vector space, ring, module, basis, cyclicity, subgroup are known concepts. In particular, the fact that any cyclic group of order $n$ is isomorphic to $\mathbb{Z}/n\mathbb{Z}$.

Throughout these lecture notes, we will make use of the following notation

- For a set $S$ we denote by $|S|$ its cardinality and by $S^C$ its complement.
- $\mathbb{Z}/n\mathbb{Z}$ denotes the integers modulo $n$.
- $\varphi(n)$ denotes the Euler-totient function.
- $\mathrm{Id}_n$ denotes the identity matrix of size $n$.
- $\mathrm{GL}_q(n)$ denotes the general linear group of degree $n$ in $\mathbb{F}_q$, i.e., all invertible matrices in $\mathbb{F}_q^{n\times n}$.
- For a matrix $M$ we write $\mathrm{rk}(M)$ to denote its rank, $\det(M)$ to denote its determinant and by $M^\top$ we denote its transpose.
- For a function $f$ we denote by $\ker(f)$ its kernel and by $\mathrm{im}(f)$ its image.
- As we can also see a matrix as a function, that is for a given matrix $M \in \mathbb{F}_q^{k\times n}$ we consider the function
$$f_M : \mathbb{F}_q^k \to \mathbb{F}_q^n, \quad a \mapsto aM,$$
we denote by $\ker(M)$ the kernel of this function and by $\mathrm{im}(M)$ the image of this function.
- For a vector $v \in \mathbb{F}_q^k$ and $i \in \{1,\dots,k\}$ we denote by $v_i$ the $i$th entry of the vector $v$. For a subset $S \subset \{1,\dots,k\}$ of size $s$, $v_S \in \mathbb{F}_q^s$ denotes the vector consisting of the entries of $v$ indexed by $S$.
- Similarly for a matrix: for a matrix $M \in \mathbb{F}_q^{k\times n}$ and $i \in \{1,\dots,k\}, j \in \{1,\dots,n\}$ we denote by $M_{i,j}$ the entry of $M$ in the $i$th row and $j$th column. For a subset $S \subset \{1,\dots,n\}$ of size $s$, $M_S \in \mathbb{F}_q^{k\times s}$ denotes the matrix consisting of the columns of $M$ indexed by $S$.
- For a matrix $M \in \mathbb{F}_q^{k\times n}$, we denote by $\langle M\rangle \subseteq \mathbb{F}_q^n$ the span of the rows of $M$, that is $\mathrm{im}(M) = \langle M\rangle$.

This list might get updated as we progress in the course.

# 1 Finite Fields

In this chapter we introduce finite fields and their main properties, which are useful for this lecture.

## 1.1 Summary

If you are already familiar with finite fields, here is a short summary of results, which prove to be useful for the course:

- For every prime $p$ there is a unique finite field $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ (up to isomorphism) of size $p$.
- For every prime $p$ and positive integer $m$ there is a unique finite field $\mathbb{F}_{p^m}$ (up to isomorphism) of size $p^m$. The subfield $\mathbb{F}_p$ is called the base field.
- The finite field $\mathbb{F}_{p^m}$ is a $\mathbb{F}_p$-linear vector space of dimension $m$ over $\mathbb{F}_p$.
- The set $\mathbb{F}_{p^m}^\star = \mathbb{F}_{p^m} \setminus \{0\}$ is a cyclic multiplicative group.
- In any finite field $\mathbb{F}_{p^m}$ there exist $\varphi(p^m - 1)$ many primitive elements $\alpha$, i.e.,

$$\mathbb{F}_{p^m}^\star = \{\alpha^0, \alpha^1, \ldots, \alpha^{p^m-2}\}.$$

- Any $x \in \mathbb{F}_{p^m}$ is such that $x^{p^m} = x$.
- For any $x \in \mathbb{F}_{p^m}$ with $x^p = x$, we have that $x \in \mathbb{F}_p$.
- For any prime $p$ and positive integers $m$ and $\ell$ such that $\ell \mid m$, the finite field $\mathbb{F}_{p^{m/\ell}}$ is a subfield of $\mathbb{F}_{p^m}$.
- For any finite field $\mathbb{F}_{p^m}$ the characteristic is $p$, i.e., for any $x \in \mathbb{F}_{p^m}$ we have $px = 0$.
- For any finite field $\mathbb{F}_{p^m}$, Freshman's dream allows us to do the following

$$(x + y)^p = x^p + y^p.$$

- If $\alpha$ is a primitive element in $\mathbb{F}_{p^m}$, then $\{1, \alpha, \ldots, \alpha^{m-1}\}$ builds a basis of $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. That is, any $x \in \mathbb{F}_{p^m}$ can be written as

$$x = \sum_{i=0}^{m-1} x_i \alpha^i.$$

- For any basis $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ of $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$, we can define the expansion map

$$\exp_\Gamma : \mathbb{F}_{p^m} \to \mathbb{F}_p^m, \quad x = \sum_{i=1}^{m} x_i \gamma_i \mapsto \exp_\Gamma(x) = (x_1, \ldots, x_m).$$

## 1.2 Prime Fields

**Definition 1.1.** A *finite field* is a field which is finite in size.

A first example is the prime field $\mathbb{F}_p$, which we know from the course Discrete Structures (or Elementary Number Theory).

**Theorem 1.2** (Prime Field)**.** *For every prime $p$ the integer residue ring $(\mathbb{Z}/p\mathbb{Z}, +, \cdot)$ is a field, denoted by $\mathbb{F}_p$.*

**Exercise 1.3.** *Prove Theorem 1.2 by showing that $(\mathbb{Z}/p\mathbb{Z}, +, \cdot)$ is a commutative ring with $1 \neq 0$ and any element in $\mathbb{F}_p^\star = \mathbb{F}_p \setminus \{0\}$ has a multiplicative inverse.*

**Example 1.4.** *The finite field of order 3 is given by $\mathbb{F}_3 = \mathbb{Z}/3\mathbb{Z} = \{0, 1, 2\}$.*

We now show that $\mathbb{F}_p$ is essentially the only field with $p$ elements, that is all fields with $p$ elements are isomorphic.

**Definition 1.5.** Let $(\mathbb{F}, +_\mathbb{F}, \cdot_\mathbb{F}), (\mathbb{G}, +_\mathbb{G}, \cdot_\mathbb{G})$, be two fields with additive identities $0_\mathbb{F}$, respectively $0_\mathbb{G}$ and multiplicative identities $1_\mathbb{F}$, respectively $1_\mathbb{G}$. The fields $\mathbb{F}$ and $\mathbb{G}$ are called *isomorphic*, if there exists a bijection $f : \mathbb{F} \to \mathbb{G}$, which is also a field homomorphism, that is for all $a, b \in \mathbb{F}$ we have

$$f(a +_\mathbb{F} b) = f(a) +_\mathbb{G} f(b), \quad \text{and} \quad f(a \cdot_\mathbb{F} b) = f(a) \cdot_\mathbb{G} f(b)$$

and $f(0_\mathbb{F}) = 0_\mathbb{G}$, $f(1_\mathbb{F}) = 1_\mathbb{G}$.

Note that any field homomorphism $f$ is injective, as the only ideals in $\mathbb{F}$ are $\{0_\mathbb{F}\}$ and $\mathbb{F}$. As we also have $f(1_\mathbb{F}) = 1_\mathbb{G}$, we must get $\text{Ker}(f) = \{0_\mathbb{F}\}$ and thus $f$ is injective.

If $|\mathbb{F}| = |\mathbb{G}|$, we thus also get that $f$ is surjective and in turn a field isomorphism.

**Theorem 1.6.** *Every finite field $\mathbb{F}$ with $|\mathbb{F}| = p$, for $p$ a prime, is isomorphic to $\mathbb{F}_p$.*

*Proof.* Let $(\mathbb{F}, +_\mathbb{F}, \cdot_\mathbb{F})$ be a field with $p$ elements, and denote its additive identity by $0_\mathbb{F}$ and its multiplicative identity by $1_\mathbb{F}$. Recall that it is enough to find a field homomorphism $f : \mathbb{F} \to \mathbb{F}_p$.

Consider the additive cyclic subgroup $S(1_\mathbb{F}) = \{1_\mathbb{F} +_\mathbb{F} \cdots +_\mathbb{F} 1_\mathbb{F}\}$. By Lagrange's Theorem, the order of this subgroup must divide $|\mathbb{F}| = p$, i.e., is either 1 or $p$. However, order 1 would imply $1_\mathbb{F} +_\mathbb{F} 1_\mathbb{F} = 1_\mathbb{F}$ and hence, from $1_\mathbb{F} = 0_\mathbb{F}$, we get a contradiction.

Thus, the additive cyclic subgroup $S(1_\mathbb{F}) = \mathbb{F}$ and we have

$$f : \mathbb{F} \to \mathbb{Z}/p\mathbb{Z}, \quad \underbrace{1_\mathbb{F} +_\mathbb{F} \cdots +_\mathbb{F} 1_\mathbb{F}}_{x} \mapsto x \mod p.$$

It is easy to check that

$$f(\underbrace{(1_\mathbb{F} +_\mathbb{F} \cdots +_\mathbb{F} 1_\mathbb{F})}_{x} +_\mathbb{F} \underbrace{(1_\mathbb{F} +_\mathbb{F} \cdots +_\mathbb{F} 1_\mathbb{F})}_{y}) = f(\underbrace{1_\mathbb{F} +_\mathbb{F} \cdots +_\mathbb{F} 1_\mathbb{F}}_{x+y}) = x+y = f(x)+f(y) \mod p$$

and clearly $f(0_{\mathbb{F}}) = 0 \mod p$.

We are left with showing that this correspondence extends also for multiplication.

Since

$$f((\underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{x}) \cdot_{\mathbb{F}} (\underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{y})) = f(\underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{xy}) = xy = f(x)f(y) \mod p$$

and clearly $f(1_{\mathbb{F}}) = 1 \mod p$ we have an isomorphism. □

Not every integer residue ring $\mathbb{Z}/n\mathbb{Z}$ defines a finite field, as we might have $ab = 0 \mod n$, for $a, b \neq 0 \mod n$ and thus not all non-zero elements have a multiplicative inverse.

**Example 1.7.** *$\mathbb{Z}/4\mathbb{Z}$ is not a finite field, as 2 has no multiplicative inverse.*

## 1.3 Subfields and Field Extensions

Prime fields are not the only finite fields we have, in fact, we now construct a finite field for any prime power $p^m$, where $p \in \mathcal{P}$, and $m$ is a positive integer.

**Definition 1.8.** A *subfield* $\mathbb{G}$ of a field $\mathbb{F}$ is a subset $\mathbb{G} \subseteq \mathbb{F}$, which is a field under the operations of $\mathbb{F}$.

To show that $\mathbb{G}$ is a subfield of $\mathbb{F}$, it is enough to show that $0_{\mathbb{F}}, 1_{\mathbb{F}} \in \mathbb{G}$ and that $\mathbb{G}$ is closed under addition $+_{\mathbb{F}}$ and multiplication $\cdot_{\mathbb{F}}$.

**Theorem 1.9.** *Let $\mathbb{F}$ be a finite field with $q$ elements. Then there exist a subfield $\mathbb{F}_p$ of $\mathbb{F}$ which is a prime field.*

*Proof.* Since $\mathbb{F}$ is a field, it contains the additive identity $0_{\mathbb{F}}$ and the multiplicative identity $1_{\mathbb{F}}$.

We consider again $S(1_{\mathbb{F}}) = \{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}\}$ and for $n = |\langle 1_{\mathbb{F}} \rangle|$, the group isomorphism

$$f : S(1_{\mathbb{F}}) \to \mathbb{Z}/n\mathbb{Z}, \quad \underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{x} \mapsto x \mod n.$$

Let $\underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{x}, \underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{y} \neq 0_{\mathbb{F}}$ and thus $\underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{xy} \neq 0_{\mathbb{F}}$. The map $f$ should send non-zero elements to non-zero, i.e.,

$$f(\underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{xy}) = xy \neq 0 \mod n,$$

which only holds for every $x, y \neq 0 \mod n$ if $n = p$ being a prime. Thus, $S(1_{\mathbb{F}})$ is isomorphic to $\mathbb{F}_p$ and since $S(1_{\mathbb{F}}) \subseteq \mathbb{F}$, it forms a subfield with $p$ elements. □

Within this proof, we have also seen that there can only exist one subfield of prime order.

**Definition 1.10.** Let $\mathbb{F}$ be a finite field with prime subfield $\mathbb{F}_p$. Then $p$ is called the *characteristic* of $\mathbb{F}$.

In particular, for any $x \in \mathbb{F}$ of characteristic $p$, we have that $xp = 0$. Indeed, for any $x \in \mathbb{F}$ we have

$$xp = \underbrace{x +_{\mathbb{F}} \cdots +_{\mathbb{F}} x}_{p} = x(\underbrace{1_{\mathbb{F}} +_{\mathbb{F}} \cdots +_{\mathbb{F}} 1_{\mathbb{F}}}_{p}) = x \cdot 0_{\mathbb{F}} = 0_{\mathbb{F}}.$$

As any finite field $\mathbb{F}$ with characteristic $p$ has the subfield $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$, we will denote the additive inverse now simply by 0, and the multiplicative inverse by 1.

**Definition 1.11.** Let $\mathbb{F}$ be a field. A *field extension* is a field $\mathbb{K}$ and a homomorphism $\phi : \mathbb{F} \to \mathbb{K}$.

As before, any field extension $\phi$ is injective and hence $\phi(\mathbb{F})$ is a subfield of $\mathbb{K}$. Observe that the scalar multiplication

$$\mathbb{F} \times \mathbb{K} \to \mathbb{K}, \quad (\lambda, k) \mapsto \phi(\lambda)k$$

gives $\mathbb{K}$ a $\mathbb{F}$-vector space structure.

In the other direction, we have that $\mathbb{F}$ with characteristic $p$ is a field extension of $\mathbb{F}_p$. And in turn, we get that any finite field must be of prime power size.

**Lemma 1.12.** *Let $\mathbb{F}$ be a finite field, then there exist $p \in \mathcal{P}$ and $m \in \mathbb{N}$, such that $|\mathbb{F}| = p^m$.*

*Proof.* We have seen before that any finite field has a base field $\mathbb{F}_p = S(1_{\mathbb{F}})$. Since $\mathbb{F}$ is a vector space over $\mathbb{F}_p$, we have some basis $\alpha_1, \ldots, \alpha_m$ of $\mathbb{F}$ over $\mathbb{F}_p$. That is every element of $\beta \in \mathbb{F}$ can be uniquely written as

$$\beta = \sum_{i=1}^{m} \lambda_i \alpha_i,$$

with $\lambda_i \in \mathbb{F}_p$. As we have $p$ choices for the $m$ many $\lambda_i' s$, we get a total of $p^m$ many elements in $\mathbb{F}$. □

Let $\mathbb{F}$ be a finite field of order $q = p^m$, then $m$ is called the *extension degree* of $\mathbb{F}$ over $\mathbb{F}_p$.

## 1.4 Construction of Finite Fields

To construct the finite field $\mathbb{F}$ with $q = p^m$ many elements, we consider polynomials over $\mathbb{F}_p$:

$$\mathbb{F}_p[x] = \left\{ \sum_{i \geq 0} f_i x^i \mid f_i \in \mathbb{F}_p \right\}.$$

The addition and multiplication of two polynomials is performed in the usual way, taking modulo $p$ for the coefficients. Then, $\mathbb{F}_p[x]$ forms a ring and is called the polynomial ring over $\mathbb{F}_p$.

Let us recall some definitions:

- Similar to polynomials over other fields, if $f(x) = \sum_{i=0}^{m} f_i x^i$, with $f_m \neq 0$, we denote by $m = \deg(f)$, the *degree* of $f$ and say $f$ is *monic* if $f_m = 1$. We use the convention that the zero polynomial $f(x) = 0$ has degree $-\infty$.
- We say that a polynomial $g(x) \in \mathbb{F}_p[x]$ is a *divisor* of $f(x) \in \mathbb{F}_p[x]$, if there exists a polynomial $h(x) \in \mathbb{F}_p[x]$, such that $f(x) = g(x)h(x)$.
- Clearly, every polynomial is a divisor of the zero polynomial. Additionally, 1 and $f(x)$ are always divisors of $f(x)$, these divisors are called *trivial divisors*. Thus, for any non-trivial divisor $g(x)$ of $f(x)$ we must have $1 \leq \deg(g) < \deg(f)$.
- If $f(x)$ has positive degree and only trivial divisors, then $f(x)$ is said to be *irreducible*.

**Example 1.13.** *Let us consider $f(x) = x^2 + 2x + 2 \in \mathbb{F}_3[x]$. The polynomial is monic of degree 2 and irreducible.*

Assuming the existence of a irreducible polynomial $g(x) \in \mathbb{F}_p[x]$ of degree $m$, we may construct a finite field with $p^m$ elements. The proof, that an irreducible polynomial of any positive degree $m$ exists over any prime field $\mathbb{F}_p$ is omitted.

Let $g(x) \in \mathbb{F}_p[x]$ be an irreducible polynomial of degree $m$ and let us consider

$$\mathbb{F}_p[x]/\langle g(x)\rangle = \{f(x) + h(x)g(x) \mid f(x), h(x) \in \mathbb{F}_p[x]\}.$$

That is, we quotient by the ideal generated by $g(x)$ (similar to integer residue rings, setting all multiples of $g(x)$ to zero and only considering the remainder $f(x)$.) This is well-defined, as a monic polynomial $q(x)$ of degree $m$ can uniquely be written as $q(x) = g(x)h(x) + f(x)$, where $\deg(f) < m$. Thus, we may write $q(x) = f(x) \mod g(x)$.

Hence, we may identify an element $q(x) = f(x)+h(x)g(x) \in \mathbb{F}_p[x]/\langle g(x)\rangle$ with its remainder $f(x)$ and the polynomial ring modulo $g(x)$ consists of all polynomials of degree up to $m-1$:

$$\mathbb{F}_p[x]/\langle g(x)\rangle = \left\{ f(x) = \sum_{i=0}^{m-1} f_i x^i \mid f_i \in \mathbb{F}_p \right\}.$$

Let $q(x) = f(x) + h(x)g(x), r(x) = t(x) + g(x)s(x)$ with $\deg(t), \deg(f) < m$, then

$$\begin{aligned} q(x) + r(x) &= f(x) + t(x) + g(x)(h(x) + s(x)) = f(x) + t(x) \mod g(x), \\ q(x) \cdot r(x) &= f(x)t(x) + g(x)(h(x)t(x) + f(x)s(x) + g(x)s(x)h(x)) = f(x)t(x) \mod g(x). \end{aligned}$$

Hence, introducing the operations in $\mathbb{F}_p[x]/\langle g(x)\rangle$

$$f(x) + t(x) \mod g(x), \quad \text{and } f(x)t(x) \mod g(x).$$

**Example 1.14.** *Let us consider $g(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$. The polynomial $g(x)$ is clearly monic and irreducible, as it does not have any roots in $\mathbb{F}_2$. The polynomial ring modulo $g(x)$ now consists of all polynomials with coefficients in $\mathbb{F}_2$ up to degree 2:*

$$\mathbb{F}_2[x]/\langle g(x)\rangle = \left\{ f(x) = \sum_{i=0}^{2} f_i x^i \mid f_i \in \mathbb{F}_2[x] \right\}.$$

*If we add or multiply two polynomials, we reduce them again modulo $g(x)$ to obtain the resulting polynomial in $\mathbb{F}_2[x]/\langle g(x)\rangle$.*

*For example, if $f(x) = x + 1$ and $t(x) = x^2 + 1$, then*

$$f(x) + t(x) = x^2 + x, \quad f(x)t(x) = x^3 + x^2 + x + 1 = (x + 1) + x^2 + x + 1 = x^2 \mod g(x).$$

The size of $\mathbb{F}_p[x]/\langle g(x)\rangle$ is thus $p^m$, as we consider all polynomials of degree up to $m - 1$.

**Theorem 1.15.** *Let $p \in \mathcal{P}, m$ a positive integer and $g(x) \in \mathbb{F}_p[x]$ an irreducible polynomial of degree $m$. Then, $\mathbb{F}_p[x]/\langle g(x)\rangle$ is a finite field with $p^m$ elements.*

*Proof.* We have seen that

$$|\mathbb{F}_p[x]/\langle g(x)\rangle| = \left| \left\{ \sum_{i=0}^{m-1} f_i x^i \mid f_i \in \mathbb{F}_p \right\} \right| = p^m,$$

and have defined addition and multiplication modulo $g(x)$.

The multiplicative identity is given by the degree 0 polynomial $1(x) = 1$, and the additive identity by the zero polynomial $g(x) = 0 \mod g(x)$.

From the properties of the polynomial ring $\mathbb{Z}[x]$ we can easily check that $\mathbb{F}_p[x]/\langle g(x)\rangle$ is a commutative ring with identity. We are left with showing that any non-zero element has a multiplicative inverse.

Let $f(x) \neq 0$ and $s(x) \neq t(x) \mod g(x)$, then $f(x)s(x) \neq f(x)t(x) \mod g(x)$. This mainly follows, from the fact that $g(x)$ is irreducible. If $f(x)s(x) = f(x)t(x) \mod g(x)$, then there exist some polynomial $h(x)$ such that $f(x)(s(x) - t(x)) = g(x)h(x)$ and since $g(x)$ is irreducible, it can only have trivial divisors: 1 and $g(x)$. Thus, either $f(x) = 0 \mod g(x)$ or $s(x) - t(x) = 0 \mod g(x)$.

Thus, if we run through all non-zero polynomials $s(x) \in \mathbb{F}_p[x]/\langle g(x)\rangle$, the products $f(x)s(x) \in \mathbb{F}_p[x]/\langle g(x)\rangle$ will also see every non-zero polynomial in $\mathbb{F}_p[x]/\langle g(x)\rangle$, in particular also $1(x) = 1$. □

**Example 1.16.** *Let us consider $g(x) = x^2 + x + 1 \in \mathbb{F}_2[x]$. Then,*

$$\mathbb{F}_2[x]/\langle g(x)\rangle = \{0, 1, x, x + 1\}.$$

**Exercise 1.17.** *Show that $g(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$ is irreducible and give the addition and multiplication table of $\mathbb{F}_2[x]/\langle (x^3 + x + 1)\rangle$.*

| + | 0 | 1 | $x$ | $x+1$ |
|---|---|---|---|---|
| 0 | 0 | 1 | $x$ | $x+1$ |
| 1 | 1 | 0 | $x+1$ | $x$ |
| $x$ | $x$ | $x+1$ | 0 | 1 |
| $x+1$ | $x+1$ | $x$ | 1 | 0 |

**Table 1:** Addition in $\mathbb{F}_2[x]/\langle(x^2+x+1)\rangle$

| · | 0 | 1 | $x$ | $x+1$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | $x$ | $x+1$ |
| $x$ | 0 | $x$ | $x+1$ | 1 |
| $x+1$ | 0 | $x+1$ | 1 | $x$ |

**Table 2:** Multiplication in $\mathbb{F}_2[x]/\langle(x^2+x+1)\rangle$

## 1.5 Multiplicative Group

Let us consider $\mathbb{F}$ a finite field with $q$ elements. By the definition of a field, we have that $\mathbb{F}^\star = \mathbb{F} \setminus \{0\}$ is an abelian multiplicative group. We now show that this is a cyclic group.

**Definition 1.18.** Let $\mathbb{F}$ be a finite field with $q$ elements. An element $\alpha \in \mathbb{F}$ is called *primitive element*, if

$$\langle\alpha\rangle = \{\alpha^i \mid i \in \mathbb{N}\} = \mathbb{F}^\star.$$

**Exercise 1.19.** *Show that $\alpha \in \mathbb{F}$ of size $q$ is primitive if and only if ord$(\alpha) = q-1$.*

Recall that a degree $m$ polynomial can have at most $m$ factors of degree 1:

**Theorem 1.20** (Fundamental Theorem of Algebra)**.** *Over any field $\mathbb{F}$, a monic polynomial $f(x) \in \mathbb{F}[x]$ of degree $m$ can have at most $m$ roots in $\mathbb{F}$.*

*If $f(x)$ has $m$ roots $\beta_1, \ldots, \beta_m$, then the unique factorization of $f(x)$ is given by*

$$f(x) = \prod_{i=1}^{m}(x-\beta_i).$$

In general, any non-zero polynomial of degree $\leq d$ has $\leq d$ roots. Note that this theorem is also very useful when trying to prove that a polynomial is zero, by showing that it has more than $d$ roots.

**Theorem 1.21.** *Let $\mathbb{F}$ be a finite field, then $\mathbb{F}^\star$ is a cyclic multiplicative subgroup.*

*Proof.* To show that $\mathbb{F}^\star$ is cyclic multiplicative group, we have to show that every finite field $\mathbb{F}$ has a primitive element.

Let $\mathbb{F}$ be a finite field with $q$ elements. Clearly, for any element $\beta \in \mathbb{F}^\star$, the cyclic multiplicative subgroup $\langle\beta\rangle = \{\beta^i \mid i \in \mathbb{N}\} \subset \mathbb{F}^\star$ must have size $s = |\langle\beta\rangle| = \text{ord}(\beta)$ which is such that $s \mid (q-1)$. If $s = q-1$, we would be done, thus we assume that for all $\beta \in \mathbb{F}^\star$, we have $\text{ord}(\beta) < q-1$.

Since any $a \in \langle\beta\rangle$ is such that $a^s = 1$, we have at least $s$ elements in $\mathbb{F}$ with $a^s = 1$. Additionally, we have at most $s$ roots of $x^s - 1$ in $\mathbb{F}$, and thus we get that

$$\langle\beta\rangle = \{a \in \mathbb{F} \mid a^s = 1\}$$

and there is exactly one cyclic subgroup of order $s$.

Recall that in a cyclic group $\langle\beta\rangle$, we have $\varphi(d)$ elements of order $d$, for $d \mid s$. Thus, we have $\varphi(s)$ elements in $\langle\beta\rangle$ of order $s$. Going through all $\beta \in \mathbb{F}^\star$, we get that the number of elements in $\mathbb{F}^\star$ of order $< q-1$, is at most

$$\sum_{d|(q-1), d<q-1} \varphi(d) = \sum_{d|(q-1)} \varphi(d) - \varphi(q-1) = q - 1 - \varphi(q-1) < q-1.$$

Thus, there must exist at least one element in $\mathbb{F}$ of order $q-1$, i.e., a primitive element.

Even more, as we also have at most $\varphi(q-1)$ elements of order $q-1$, we get exactly $\varphi(q-1)$ many primitive elements.

□

**Exercise 1.22.** *Give an alternative proof that the multiplicative group is cyclic, using the fundamental theorem of finite abelian groups.*

**Example 1.23.** *In $\mathbb{F}_5$, we have $2$ is a primitive element, as $\mathbb{F}_5^\star = \{2^0, 2^1, 2^3, 2^2\}$.*

Once we are given a primitive element, we can also construct all others and any element of order $d \mid (q-1)$.

**Exercise 1.24.** *Let $\mathbb{F}$ be a finite field with $q$ elements and $\alpha \in \mathbb{F}$ be a primitive element. Show that*

- *$\alpha^\ell$ is primitive if and only if gcd$(\ell, q-1) = 1$.*
- *If $\ell \mid (q-1)$, then $\alpha^{(q-1)/\ell}$ has order $\ell$.*

## 1.6 Uniqueness

Recall that for any $\beta \in \mathbb{F}_p$ we have that $\beta^p = \beta$ and hence

$$x^p - x = \prod_{\beta \in \mathbb{F}_p} (x - \beta).$$

If $|\mathbb{F}^\star| = q-1$, then by Lagrange's Theorem we have that $\beta^{q-1} = 1$ and hence every element of $\mathbb{F}^\star$ is a root of the polynomial $x^{q-1} - 1$. To include the zero element we simply consider

$$x(x^{q-1} - 1) = x^q - x.$$

As a polynomial of degree $q$ can have at most $q$ roots in $\mathbb{F}$, we get that

$$x^q - x = \prod_{\beta \in \mathbb{F}} (x - \beta).$$

**Theorem 1.25.** *Every finite field $\mathbb{F}$ with $q = p^m$ elements is isomorphic to $\mathbb{F}_p[x]/\langle g(x) \rangle$, where $g(x) \in \mathbb{F}_p[x]$ is an irreducible polynomial of degree $m$.*

In order to prove this result, we resort to the definition of splitting fields.

**Definition 1.26.** Let $\mathbb{F}$ be a field and $f(x) \in \mathbb{F}[x]$. A *splitting field* of $f(x)$ over $\mathbb{F}$ is a field extension $\mathbb{K}$, such that

$$f(x) = c \prod_{i=1}^{n} (x - \alpha_i),$$

for $\alpha_i \in \mathbb{K}$ and $c \in \mathbb{F}$ and $f(x)$ does not split in a proper subfield of $\mathbb{K}$.

In short, the splitting field of $f(x)$ is the smallest field $\mathbb{K}$ where $f(x)$ splits into linear factors, i..e, $\mathbb{K}$ contains all roots of $f(x)$. We note that splitting fields always exist and they are unique (up to isomorphism).

Note that if $\mathbb{F}$ is a field with $q$ elements and $\mathbb{L}$ is any subfield of $\mathbb{F}$, then $\mathbb{F}$ is the splitting field of $x^q - x$ over $\mathbb{L}$.

Hence, by the uniqueness of splitting fields (up to isomorphisms) we also get that $\mathbb{F}$ with $q = p^m$ elements is unique, and from now on denoted by $\mathbb{F}_q$ or $\mathbb{F}_{p^m}$ to emphasize the base field $\mathbb{F}_p$.

## 1.7 Different Representation

Instead of considering elements in the finite field $\mathbb{F}_{p^m}$ as polynomials over $\mathbb{F}_p$ modulo an irreducible polynomial $f(x)$ of degree $m$, we may always use a root $\alpha$ of $f(x)$ to represent the elements of $\mathbb{F}_{p^m}$.

**Definition 1.27.** Let $p$ be a prime and $m$ a positive integer. Let $f(x) = \sum_{i=0}^{m-1} f_i x^i + x^m \in \mathbb{F}_p[x]$ be an irreducible polynomial of degree $m$ and let $\alpha$ be a root of $f(x)$. Then, $\mathbb{F}_p$ *adjoin* $\alpha$ is

$$\mathbb{F}_p(\alpha) = \{ \sum_{i=0}^{m-1} a_i \alpha^i \mid a_i \in \mathbb{F}_p \}.$$

The usual definition is a bit more abstract, stating that for $\alpha \in \mathbb{L}$ :

$$\mathbb{F}_p(\alpha) = \bigcap_{\mathbb{F}_p \subset \mathbb{K} \subset \mathbb{L}, \alpha \in \mathbb{K}} \mathbb{K}.$$

Note that $\mathbb{F}_p(\alpha)$ is the smallest field extension of $\mathbb{F}_p$ containing $\alpha$. The fact that $\alpha$ is contained in $\mathbb{F}_p(\alpha)$ is clear, as we can choose

$$a_1 = 1, \quad a_i = 0 \quad \forall\, i \neq 1.$$

Similarly, we can consider all $a_0 \in \mathbb{F}_p$ and $a_i = 0$ for all $i > 0$ to show that $\mathbb{F}_p \subset \mathbb{F}_p(\alpha)$. Also the other properties can be easily checked.

**Theorem 1.28.** *Let $p$ be a prime and $m$ a positive integer. Let $f(x) = \sum_{i=0}^{m-1} f_i x^i + x^m \in \mathbb{F}_p[x]$ be an irreducible polynomial of degree $m$ and let $\alpha$ be a root of $f(x)$. Then,*

$$\mathbb{F}_p[x]/\langle f(x)\rangle \cong \mathbb{F}_p(\alpha).$$

*Proof.* Let us consider the map

$$\begin{aligned} \text{ev}_\alpha : \mathbb{F}_p[x]/\langle f(x)\rangle &\to \mathbb{F}_p(\alpha), \\ a(x) = \sum_{i=0}^{m-1} a_i x^i &\mapsto a = \sum_{i=0}^{m-1} a_i \alpha^i, \end{aligned}$$

i.e., sending $a(x)$ to $a(\alpha)$. This is clearly a homomorphism, as

$$\begin{aligned} \text{ev}_\alpha(a(x) + b(x)) &= \text{ev}_\alpha(a(x)) + \text{ev}_\alpha(b(x)), \\ \text{ev}_\alpha(a(x)b(x)) &= \text{ev}_\alpha(a(x))\text{ev}_\alpha(b(x)). \end{aligned}$$

Since $f(x)$ is irreducible, we have that

$$\ker(\text{ev}_\alpha) = \langle f(x)\rangle$$

and by the first isomorphism theorem, we get

$$\mathbb{F}_p[x]/\langle f(x)\rangle \cong \text{im}(\text{ev}_\alpha).$$

Clearly, $\text{im}(\text{ev}_\alpha)$ contains $\mathbb{F}_p$ and $\alpha$ and as it is isomorphic to $\mathbb{F}_p[x]/\langle f(x)\rangle$ it is also a field, inside the smallest field containing $\mathbb{F}_p$ and $\alpha$. Hence, they must be isomorphic. □

Instead of considering any irreducible polynomial, one can also consider a primitive polynomial.

**Definition 1.29.** Let $\alpha \in \mathbb{F}_{p^m}$ be a primitive element. The minimal polynomial of $\alpha$ is called *primitive polynomial.*

To check whether an irreducible polynomial $f(x)$ of degree $m$ is primitive, we can equivalently check if the smallest positive integer $n$ such that $f(x)$ divides $x^n - 1$ is $n = p^m - 1$.

**Example 1.30.** *Let us consider $\mathbb{F}_3$. The polynomial $x^2 + 1$ is irreducible, but not primitive as it divides $x^4 - 1$. A primitive polynomial of degree 2 would for example be $x^2 + 2x + 2$.*

Once we have identified a primitive polynomial $f(x) \in \mathbb{F}_p[x]$ and a root $\alpha$, we can immediately write all elements of $\mathbb{F}^\star_{p^m}$ as powers of $\alpha$.

**Example 1.31.** *We are allowed to use any irreducible polynomial, as*

$$\mathbb{F}_9 \cong \mathbb{F}_3[x]/\langle x^2 + 1\rangle \cong \mathbb{F}_3[x]/\langle x^2 + 2x + 2\rangle.$$

*Let us denote by $\alpha$ a root of $x^2 + 2x + 2$, i.e., $\alpha^2 = \alpha + 1$ and $\beta$ a root of $x^2 + 1$, i.e., $\beta^2 = 2$. While we can write*

$$\mathbb{F}_9 \cong \mathbb{F}_3(\alpha) \cong \mathbb{F}_3(\beta),$$

*only $\alpha$ generates the multiplicative group $\mathbb{F}^\star_9$:*

$$\alpha^0 = 1, \alpha^1 = \alpha, \alpha^2 = \alpha + 1, \alpha^3 = 2\alpha + 1, \alpha^4 = 2, \alpha^5 = 2\alpha, \alpha^6 = 2\alpha + 2, \alpha^7 = \alpha + 2.$$

**Exercise 1.32.** *Give the multiplication table of* $\mathbb{F}_4 \cong \mathbb{F}_2(\alpha)$*, where* $\alpha$ *is a root of* $x^2 + x + 1$.

We can also use a different representation, as $\mathbb{F}_{p^m} \cong \mathbb{F}_p^m$ as $\mathbb{F}_p$- vector space.
For this we define the expansion map.

**Definition 1.33.** Let $\Gamma = \{\gamma_0, \ldots, \gamma_{m-1}\}$ be a basis of $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. Then the expansion map with respect to $\Gamma$ is given by

$$\begin{aligned} \exp_\Gamma : \mathbb{F}_{p^m} &\to \mathbb{F}_p^m, \\ a = \sum_{i=0}^{m-1} a_i \gamma_i &\mapsto \exp_\Gamma(a) = (a_0, \ldots, a_{m-1}). \end{aligned}$$

This map is $\mathbb{F}_p$-linear, meaning that

- For $a, b \in \mathbb{F}_{p^m}$ we have $\exp_\Gamma(a + b) = \exp_\Gamma(a) + \exp_\Gamma(b)$,
- for $a \in \mathbb{F}_{p^m}$ and $\lambda \in \mathbb{F}_p$ we have that $\exp_\Gamma(\lambda a) = \lambda \exp_\Gamma(a)$.

As we also want to handle $\exp_\Gamma(ab)$ we need to introduce the multiplication matrix.

For this we will focus on a basis $\Gamma = \{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$ of $\mathbb{F}_p^m$ over $\mathbb{F}_p$, we call such a basis a *polynomial basis*. Note that $\{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$ is a basis of $\mathbb{F}_p^m$ over $\mathbb{F}_p$, if and only if $\alpha$ is a root of $f(x) \in \mathbb{F}_p[x]$, which is irreducible and of degree $m$.

Let $\alpha$ be the root of an irreducible polynomial $f(x) = \sum_{i=0}^{m-1} f_i x^i + x^m \in \mathbb{F}_p[x]$ of degree $m$ and consider

$$\begin{array}{lll} \mathbb{F}_p[x]/\langle f(x)\rangle & \overset{\text{ev}_\alpha}{\cong} \mathbb{F}_p(\alpha) & \overset{\exp_\Gamma}{\cong} \mathbb{F}_p^m, \\ a(x) = \sum_{i=0}^{m-1} a_i x^i & \mapsto a = \sum_{i=0}^{m-1} a_i \alpha^i & \mapsto \exp_\Gamma(a) = (a_0, \ldots, a_{m-1}), \\ a(x) + b(x) & \mapsto a + b & \mapsto \exp_\Gamma(a) + \exp_\Gamma(b). \end{array}$$

However, what happens to the multiplication? We can compute $a(x)b(x) \mod f(x)$ and $ab \in \mathbb{F}_p(\alpha)$, thus we need to figure out what $\exp_\Gamma(ab)$ is in terms of $\exp_\Gamma(a)$ and $\exp_\Gamma(b)$.

For this we define the multiplication matrix for $b \in \mathbb{F}_p(\alpha)$ via the basis $\Gamma$ as

$$M_\Gamma(b) = \begin{pmatrix} \exp_\Gamma(b) \\ \exp_\Gamma(\alpha b) \\ \vdots \\ \exp_\Gamma(\alpha^{m-1} b) \end{pmatrix}$$

and define

$$\exp_\Gamma(a) \circ \exp_\Gamma(b) = \exp_\Gamma(a) M_\Gamma(b).$$

Note that for $b = \alpha$, the multiplication matrix $M_\Gamma(\alpha)$ is the companion matrix of $f(x)$ as

$$M_\Gamma(\alpha) = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -f_0 & -f_1 & -f_2 & \cdots & -f_{m-1} \end{pmatrix}.$$

**Example 1.34.** *Let us consider $\mathbb{F}_8 \cong \mathbb{F}_2(\alpha)$ where $\alpha$ is a primitive root and satisfies $\alpha^3 = \alpha + 1$ as the primitive polynomial over $\mathbb{F}_2$ is given by $x^3 + x + 1$. Thus, we have the polynomial basis $\Gamma = \{1, \alpha, \alpha^2\}$ of $\mathbb{F}_8$ over $\mathbb{F}_2$. Let $a = \alpha^2 + 1$ and $b = \alpha + 1$. We can easily expand them to $\mathbb{F}_2^3$ as*

$$exp_\Gamma(a) = (1, 0, 1), \quad exp_\Gamma(b) = (1, 1, 0).$$

*We want to multiply $a$ with $b$ and over $\mathbb{F}_8$ we can easily check that $ab = \alpha^2$ which is expanded to*

$$exp_\Gamma(ab) = (0, 0, 1).$$

*We compute*

$$M_\Gamma(b) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

*and check*

$$M_\Gamma(b) exp_\Gamma(a) = (0, 0, 1) = exp_\Gamma(ab).$$

To summarize, we may view the finite field $\mathbb{F}_{p^m}$ in three different ways:

Let $p$ be a prime and $m$ a positive integer. Let $f(x) = \sum_{i=0}^{m-1} f_i x^i + x^m \in \mathbb{F}_p[x]$ be an irreducible polynomial of degree $m$ and let $\alpha$ be a root of $f(x)$.

| $\mathbb{F}_p[x]/\langle f(x)\rangle$ | $\mathbb{F}_p(\alpha)$ | $\mathbb{F}_p^m$ |
|---|---|---|
| $a(x) = \sum_{i=0}^{m-1} a_i x^i$ | $a = \sum_{i=0}^{m-1} a_i \alpha^i$ | $(a_0, \ldots, a_{m-1})$ |
| $a(x) + b(x) \mod f(x)$ | $a + b$ | $\exp_\Gamma(a) + \exp_\Gamma(b)$ |
| $a(x) \cdot b(x) \mod f(x)$ | $a \cdot b$ | $\exp_\Gamma(a) \circ \exp_\Gamma(b)$ |

**Example 1.35.** *Let us consider again*

$$\mathbb{F}_9 \cong \mathbb{F}_3[x]/\langle x^2 + 1\rangle \cong \mathbb{F}_3[x]/\langle x^2 + 2x + 2\rangle,$$

*and denote by $\alpha$ a root of $x^2 + 2x + 2$, i.e., $\alpha^2 = \alpha + 1$ and $\beta$ a root of $x^2 + 1$, i.e., $\beta^2 = 2$.*

*Recall that for $\alpha$ a primitive element over $\mathbb{F}_q$ and $\beta$ of order $z$, we have that $\beta = \alpha^{i(q-1)/z}$ for some $i \in \{1, \ldots, z\}$ and gcd$(i, (q-1)/z) = 1$. Thus, assuming $i = 1$, we get that $\beta = \alpha^2 = \alpha + 1$, and can convert the different representations:*

| $\mathbb{F}_3(\beta)$ | 0 | 1 | 2 | $\beta$ | $2\beta$ | $\beta+1$ | $\beta+2$ | $2\beta+1$ | $2\beta+2$ |
|---|---|---|---|---|---|---|---|---|---|
| $\mathbb{F}_3(\alpha)$ | 0 | 1 | 2 | $\alpha+1$ | $2\alpha+2$ | $\alpha+2$ | $\alpha$ | $2\alpha$ | $2\alpha+1$ |
| $\langle\alpha\rangle$ | | $\alpha^0$ | $\alpha^4$ | $\alpha^2$ | $\alpha^6$ | $\alpha^7$ | $\alpha^1$ | $\alpha^5$ | $\alpha^3$ |

## 1.8 Trace and Norm

We define two more important maps: the trace and the norm.

**Definition 1.36.** An *automorphism* $\sigma$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is a ring homomorphism from $\mathbb{F}_{q^m}$ to itself such that $\sigma_{\mathbb{F}_q} = \mathrm{id}$ is the identity.

Note that it follows from the definition that such automorphism is bijective. Consider the map

$$\sigma_j : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}, \quad \alpha \mapsto \alpha^{q^j}.$$

**Theorem 1.37.** *Let $\mathbb{F}_{q^m}$ be a finite field with $q^m$ elements ($q$ is a prime power). The automorphisms of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ are the distinct maps $\sigma_0, \sigma_1, \ldots, \sigma_{m-1}$.*

*Proof.* Clearly, any $\sigma_j$ is a ring homomorphism. Moreover, as any element in $\mathbb{F}_q$ satisfies $\alpha^q = \alpha$, we have that $\sigma_j$ fixes the elements of $\mathbb{F}_q$. In order to see that the $\sigma_j$ are distinct, note that if $\beta$ is a primitive element in $\mathbb{F}_{q^m}$, then $\beta^{q^i} = \beta^{q^j}$ forces $q^i = q^j \mod q^m - 1$, and thus $i = j \in \{1, \ldots, m-1\}$.

We are left with showing that there are no other automorphisms. Let $\sigma$ be an automorphism of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and $\beta$ be a primitive element in $\mathbb{F}_q$ and

$$f(x) = x^m + f_{m-1}x^{m-1} + \cdots + f_0 \in \mathbb{F}_q[x]$$

be its minimal polynomial over $\mathbb{F}_q$. Then,

$$\begin{aligned} 0 =& \sigma(\beta^m + f_{m-1}\beta^{m-1} + \cdots + f_0) \\ =& \sigma(\beta)^m + f_{m-1}\sigma(\beta)^{m-1} + \cdots + f_0, \end{aligned}$$

thus $\sigma(\beta)$ is a root of $f(x)$ in $\mathbb{F}_{q^m}$. We now show that any root of a primitive polynomial $f(x) = \sum_{i=0}^{m-1} f_i x^i + x^m$ is of the form $\beta^{q^j}$ for some $j \in \{0, \ldots, m-1\}$. Note that it is enough to show that since $\beta$ is a root, so is $\beta^q$, as then we can repeat this argument for $\beta^{q^j}$ to get the root $\beta^{q^{j+1}}$.

In fact, since $\beta$ is a root we have that

$$f(\beta) = \sum_{i=0}^{m-1} f_i \beta^i + \beta^m = 0.$$

Now

$$\begin{aligned} f(\beta^q) &= \sum_{i=}^{m-1} f_i(\beta^q)^i + (\beta^q)^m \\ &= \sum_{i=}^{m-1} f_i^q \beta^{qi} + \beta^{qm} \\ &= \sum_{i=}^{m-1} (f_i\beta^i)^q + (\beta^m)^q \\ &= (\sum_{i=}^{m-1} f_i\beta^i + \beta^m)^q = (f(\beta))^q = 0, \end{aligned}$$

or by simply observing that $\sigma$ is a automorphism. Clearly, there are not more roots, as the degree of $f(x)$ is $m$.

This implies that $\sigma(\beta) = \beta^{q^j}$ for some $j \in \{0, \ldots, m-1\}$, thus $\sigma = \sigma_j$. □

**Corollary 1.38.** *The group of automorphisms of $\mathbb{F}_{q^m}$ is a cyclic group of order $m$, generated by $\sigma_1$.*

This map $\sigma_1 : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}, \alpha \mapsto \alpha^q$ is called *Frobenius map.*

**Definition 1.39.** Let $\mathbb{F}_{q^m}$ and $\mathbb{F}_q$ be the finite field with $q^m$, respectively $q$, elements. The *trace* map is defined as

$$\begin{aligned} \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q} : \mathbb{F}_{q^m} &\to \mathbb{F}_q, \\ \alpha &\mapsto \sum_{i=0}^{m-1} \alpha^{q^i}. \end{aligned}$$

Note that if $q = p$ is prime we call the trace map $\mathrm{Tr}_{\mathbb{F}_{p^m}/\mathbb{F}_p} = \mathrm{Tr}_{\mathbb{F}_{p^m}}$ the *absolute trace*.

It might not directly be clear why this map sends elements from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$. We have a simple test to check whether an element of $x \in \mathbb{F}_{q^m}$ is actually living in the subfield $\mathbb{F}_q$: check if $x^q = x$.

Thus, we compute

$$\left(\sum_{i=0}^{m-1} \alpha^{q^i}\right)^q = \sum_{i=0}^{m-1} \alpha^{q^i \cdot q} = \sum_{i=0}^{m-1} \alpha^{q^{i+1}} = \sum_{i=0}^{m-1} \alpha^{q^i},$$

where we have used that $\alpha^{q^m} = \alpha$.

The traces possess many interesting properties, which we leave as an exercise:

**Theorem 1.40.** *Let $\alpha, \beta \in \mathbb{F}_{q^m}$ and $\lambda \in \mathbb{F}_q$. Then*

1. $Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha + \beta) = Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) + Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\beta)$,
2. $Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\lambda\alpha) = \lambda Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)$,
3. $Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\lambda) = m\lambda$,
4. $Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha^q) = Tr_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)$.

**Exercise 1.41.** *Prove Theorem 1.40.*

The next map is defined in a similar fashion:

**Definition 1.42.** Let $\mathbb{F}_{q^m}$ and $\mathbb{F}_q$ be the finite field with $q^m$, respectively $q$, elements. The *norm* map is defined as

$$\begin{aligned} N_{\mathbb{F}_{q^m}/\mathbb{F}_q} : \mathbb{F}_{q^m} &\to \mathbb{F}_q, \\ \alpha &\mapsto \prod_{i=0}^{m-1} \alpha^{q^i} = \alpha^{(q^m-1)/(q-1)}. \end{aligned}$$

We can again use our test (this time $x^{q-1} = 1$), to check whether we land in $\mathbb{F}_q$, by computing

$$\left(\alpha^{(q^m-1)/(q-1)}\right)^{q-1} = \alpha^{q^m-1} = 1.$$

We get similar properties to trace, again left as an exercise:

**Theorem 1.43.** *Let $\alpha, \beta \in \mathbb{F}_{q^m}$ and $\lambda \in \mathbb{F}_q$. Then*

1. $N_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha\beta) = N_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) N_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\beta)$,
2. $N_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\lambda) = \lambda^m$,
3. $N_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha^q) = N_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)$.

**Exercise 1.44.** *Prove Theorem 1.43.*

## 1.9 Some Properties

We may ask if $\mathbb{F}_{p^m}$ has also other subfields than $\mathbb{F}_p$.

**Theorem 1.45.** *Let $\mathbb{F}_q$ be a field with $q = p^m$ elements. $\mathbb{F}_q$ has a subfield of order $p^r$, if and only if $r \mid m$.*

*Proof.* Recall that $\mathbb{F}_{p^m}$ is a $\mathbb{F}_p$-vector space of degree $m$. Thus, for any subfield $\mathbb{F} \subset \mathbb{F}_{p^m}$, we must have that $\mathbb{F}$ has order $p^r$, where $r \mid m$. For the other direction, we assume that $r \mid m$ and construct a subfield of $\mathbb{F}_{p^m}$ with $p^r$ elements. Let $d = m/r$ and note that

$$p^m - 1 = (p^r - 1)(1 + p^r + \cdots + p^{r(d-1)}).$$

Thus, $p^r - 1$ divides $p^m - 1$. Similarly, we see that

$$(x^{p^r-1} - 1) \mid (x^{p^m-1} - 1).$$

Since $\mathbb{F}_{p^m}$ is the splitting fields of $x^{p^m} - x = x(x^{p^m-1} - 1)$ over $\mathbb{F}_p$, it contains all roots of $x^{p^r-1} - 1$. Together with 0, these roots form a subfield of cardinality $p^r$, as claimed. □

And we conclude this short recap on finite fields with Freshman's dream.

**Theorem 1.46** (Freshman's dream)**.** *For any $a, b \in \mathbb{F}_{p^m}$ we have that*

$$(a + b)^p = a^p + b^p.$$

*Proof.* By the binomial theorem we have that

$$(a + b)^p = \sum_{i=0}^{p} \binom{p}{i} a^i b^{p-i}.$$

Since for all $i \in \{1, \ldots, p - 1\}$ we have $p \mid \binom{p}{i} = \frac{p!}{i!(p-i)!}$, we get that $\binom{p}{i} a^i b^{p-i} = 0$ in $\mathbb{F}_{p^m}$. The only non-zero terms in this sum are $i = 0$ and $i = p$, where $\binom{p}{0} = \binom{p}{p} = 1$. □

## 1.10 Invertible Matrices

Finally, we may consider the invertible matrices.

**Proposition 1.47.** *Let $q$ be a prime power and $n$ a positive integer. Then*

$$|GL_q(n)| = \prod_{i=0}^{n-1} (q^n - q^i).$$

*Proof.* To count the invertible matrices in $\mathbb{F}_q^{n \times n}$, we may start with any non-zero vector $r_1 \in \mathbb{F}_q^n$ and use this as first row. Clearly, we have $q^n - 1$ choices. For the second row, we may choose any vector, which is not a multiple of $r_1$, i.e., $r_2 \in \mathbb{F}_q^n \setminus \langle r_1 \rangle$. Clearly, we have $q^n - q$ choices. We continue this way, for the $i$th row choosing $r_i \in \mathbb{F}_q^n \setminus \langle r_1, \ldots, r_{i-1} \rangle$, for which we have $q^n - q^{i-1}$ choices. □

**Exercise 1.48.** *Let $q$ be a prime power and $m < n$ be positive integers. Perform a similar counting argument, to show that the number of $m \times n$ matrices over $\mathbb{F}_q$ of full rank $m$ is given by*

$$\prod_{i=0}^{m-1} (q^n - q^i).$$

**Corollary 1.49.** *Let $q$ be a prime power and $m < n$ be positive integers. The probability for a uniformly at random chosen matrix $A \in \mathbb{F}_q^{m \times n}$ to have full rank is*

$$\prod_{i=n-m+1}^{n} (1 - q^{-i}).$$

# 2 Basics of Codes

As we have seen in our introductory example, we would like to *encode* a message, i.e., add redundancy to it, before sending it through a channel prone to errors.

The encoding in our example, was given by repeating the message three times. This ensured the message recovery if one error happens. In fact, any sent vector must be of the form $(x, x, x)$, for some $x \in \mathbb{F}_2^3$. If only one error happens in the channel, it will only effect one of the three $x$'s and as soon as we receive $(y, x, x), (x, y, x)$ or $(x, x, y)$, we can easily recover the sent message $x$. The sent vectors are now much larger, of length $3 \cdot 3$, in our example. However, we could also give a different encoding, which is still able to recover the message after one error was inserted, but with a much shorter length:

Let us consider again $m \in \mathbb{F}_2^3$ and a channel which inserts one error. By encoding $c = mG$, where

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

we get that the sent vectors must live in the set

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

Thus, if one error happens, i.e., we receive $c + e$, there is no other $c' \in \mathcal{C}$ such that $c + e = c' + e'$.

As you can see from our toy example, $\mathcal{C}$, being the rowspan of $G$, enjoys linearity, i.e., for any $c, c' \in \mathcal{C}$ also $\lambda c + \mu c' \in \mathcal{C}$, for $\lambda, \mu \in \mathbb{F}_2$ scalars.

Thus, it might not come as a surprise, that our main objects for this lecture, called codes, are defined as linear subspace $\mathcal{C} \subset \mathbb{F}_q^n$.

Let us formalize the concepts we have observed above, to get a well-founded theory, of what we mean with encoding, decoding, error-correction capability and so on.

## 2.1 Generator Matrix and Parity-Check Matrix

Let us fix that $\mathbb{F}_q$ will denote the finite field of $q$ elements, where $q$ is a prime power.

**Definition 2.1** (Linear Code)**.** Let $1 \leq k \leq n$ be integers. Then, an $[n, k]_q$ *linear code* $\mathcal{C}$ over $\mathbb{F}_q$ is a $k$-dimensional linear subspace of $\mathbb{F}_q^n$.

Note that we emphasize the linearity, as a *code* is simply any subset $\mathcal{C} \subseteq \mathbb{F}_q^n$.

We have a certain terminology in coding theory, which we will continuously update. The first terms are

Let $\mathcal{C}$ be an $[n,k]_q$ linear code.

- The parameter $n$ is called the *length* of the code.
- The parameter $k$ is called the *dimension* of the code.
- The elements in the code are called *codewords*.
- The parameter $r = n - k$ is called the *redundancy*.
- The parameter $R = k/n$ is called the *rate* of the code.

We say that a code is *non-degenerate* if for any $i \in \{1, \dots, n\}$ there exists some $c \in \mathcal{C}$ with $c_i \neq 0$.

As $\mathcal{C}$ is linear, it must have some basis, which allows us to represent it compactly. In fact, linear codes allow for an easy representation through their *generator matrices*, which have the code as an image.

**Definition 2.2** (Generator Matrix)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n,k]_q$ linear code. Then, a matrix $G \in \mathbb{F}_q^{k \times n}$ is called a *generator matrix* of $\mathcal{C}$ if

$$\mathcal{C} = \left\{ xG \mid x \in \mathbb{F}_q^k \right\},$$

that is, the rows of $G$ form a basis of $\mathcal{C}$.

We will often write $\langle G \rangle$ to denote the code generated by the rows $G$. Thus, we can easily check if a code is degenerate by checking whether a generator matrix has a zero column.

**Example 2.3.** *A repetition code $\mathcal{C} = \langle G \rangle$, is defined through the generator matrix*

$$G = \begin{pmatrix} 1 \cdots 1 \end{pmatrix} \in \mathbb{F}_q^{1 \times n}.$$

*Thus, the code has dimension 1 and the rate is $R = \frac{1}{n}$.*

Given an $[n,k]_q$ linear code $\mathcal{C} = \langle G \rangle$, to *encode* a message $m \in \mathbb{F}_q^k$, we apply the map

$$\text{Enc} : \mathbb{F}_q^k \to \mathbb{F}_q^n, \quad m \mapsto mG.$$

Note that an $[n,k]_q$ linear code $\mathcal{C}$ has dimension $k$, that is $|\mathcal{C}| = q^k$.

Thus, a generator matrix $G \in \mathbb{F}_q^{k \times n}$ of $\mathcal{C}$ has full rank $k$ and in turn, the encoding map Enc is injective.

One can also represent a code through a matrix $H$, which has the code as kernel.

**Definition 2.4** (Parity-Check Matrix)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n,k]_q$ linear code. Then, a matrix $H \in \mathbb{F}_q^{(n-k) \times n}$ is called a *parity-check matrix* of $\mathcal{C}$, if

$$\mathcal{C} = \left\{ y \in \mathbb{F}_q^n \mid yH^\top = 0 \right\}.$$

For any $x \in \mathbb{F}_q^n$, we call $xH^\top$ the *syndrome* of $x$ through $H$.

Let $\mathcal{C}$ be an $[n,k]_q$ linear code with $\langle G\rangle = \mathcal{C} = \ker(H^\top)$.

- The parameter $n$ is called the *length* of the code.
- The parameter $k$ is called the *dimension* of the code.
- The elements in the code are called *codewords*.
- The parameter $r = n - k$ is called the *redundancy*.
- The parameter $R = k/n$ is called the *rate* of the code.
- The matrix $G$ is called a *generator matrix* of the code.
- The matrix $H$ is called a *parity-check matrix* of the code.
- The vector $s = xH^\top$ is called a *syndrome* of $x$.

The parity-check matrix gives an easy way to check whether some vector $x$ is a codeword or not, by simply computing its syndrome $xH^\top$. If the syndrome is zero $x$ is a codeword and if the syndrome is not zero, $x$ is not a codeword.

The name parity-check matrix comes from single parity-check codes.

**Example 2.5.** *In several applications, one uses a single parity-check code, i.e., adding a last digit, which serves to check whether the whole vector is a valid codeword or not. Over the binary, we are given a message* $m = (m_1, \ldots, m_k) \in \mathbb{F}_2^k$ *and we want to encode* $m$ *to* $c = (m_1, \ldots, m_k, y) \in \mathbb{F}_2^{k+1}$*, where* $y = \sum_{i=1}^k m_i$*. A generator matrix of such a single parity-check code is given by*

$$G = \begin{pmatrix} & 1 \\ Id_k & \vdots \\ & 1 \end{pmatrix}$$

*and a parity-check matrix is simply given by*

$$H = \begin{pmatrix} 1 \cdots 1 \end{pmatrix},$$

*as*

$$Hc^\top = \sum_{i=1}^{k} m_i + y \equiv 0 \mod 2.$$

Thus, the name parity-check matrix comes from the binary code, where one checks if a vector has even weight.

**Exercise 2.6.** *How would a* $[n, n-1]_q$ *single parity-check code for* $q \neq 2$ *be defined?*

Note that there are many generator matrices $G$ which have the same code as image. In fact, for any $U \in \mathrm{GL}_k(q)$ we have that $\langle UG \rangle = \langle G \rangle$, as the row operation $U$ only changes the basis but not the subspace.

Recall the rank-nullity theorem, stating that for any $A \in \mathbb{F}_q^{n \times m}$,

$$\dim(\ker(A)) = n - \mathrm{rk}(A).$$

Since $\ker(H^\top) = \mathcal{C}$ has dimension $k$ we have that $H \in \mathbb{F}_q^{(n-k) \times n}$ has full rank $n - k$.

We could think of a generator matrix and parity-check matrix as forming a short exact sequence:

$$0 \to \mathbb{F}_q^k \xrightarrow{G} \mathbb{F}_q^n \xrightarrow{H^\top} \mathbb{F}_q^{n-k} \to 0.$$

Let $H \in \mathbb{F}_q^{(n-k) \times n}$ be a parity-check matrix and assume $x \in \mathbb{F}_q^n$ is unknown. Then, we get a system of $n - k$ linear equations in $x_i$ from $Hx^\top = s^\top$:

$$\begin{aligned} \sum_{i=1}^{n} h_{1,i} x_i &= s_1 \\ &\vdots \\ \sum_{i=1}^{n} h_{n-k,i} x_i &= s_{n-k}. \end{aligned}$$

These equations are called *parity-check equations* or *syndrome equations.*

Consider our original problem, where we have received $r = c + e \in \mathbb{F}_q^n$, with $c \in \mathcal{C}$ the sent codeword and $e$ an error vector added by the channel. By computing the syndrome of $r$ via the parity-check matrix $H$ of $\mathcal{C}$, we get

$$s = rH^\top = (c + e)H^\top = cH^\top + eH^\top = eH^\top,$$

i.e., we see that the received word is erroneous, and get an equation only depending on the error vector.

Since $\mathcal{C} = \mathrm{Im}(G) = \ker(H^\top)$, we also get a relation between the two matrices, namely

$$GH^\top = 0.$$

For $x, y \in \mathbb{F}_q^n$ let us denote by $\langle x, y \rangle$ the standard inner product, i.e.,

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i.$$

Then, we can define the dual of an $[n, k]_q$ linear code $\mathcal{C}$ as the orthogonal space of $\mathcal{C}$.

**Definition 2.7** (Dual Code)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n,k]_q$ linear code. The *dual code* $\mathcal{C}^\perp$ is an $[n, n-k]_q$ linear code, defined as

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n \mid \langle x, y\rangle = 0 \; \forall \; y \in \mathcal{C}\}.$$

We have to be careful with the term "dual code": we do not intend the space of linear forms on $\mathcal{C}$, instead we say "dual" to intend the orthogonal space with respect to the standard inner product. And even though $\dim(\mathcal{C}) + \dim(\mathcal{C}^\perp) = n$, we do not necessarily have that $\mathcal{C} \cap \mathcal{C}^\perp = \{0\}$. In fact, we can even have $\mathcal{C} = \mathcal{C}^\perp$.

**Exercise 2.8.** *Show that the code generated by*

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

*is such that* $\mathcal{C} = \mathcal{C}^\perp$.

**Exercise 2.9.** *Show that a parity-check matrix of* $\mathcal{C}$ *is in fact a generator matrix of* $\mathcal{C}^\perp$.

**Example 2.10.** *Let* $\mathcal{C}$ *be the* $[n,1]_2$ *linear repetition code, generated by*

$$G = \begin{pmatrix} 1 \cdots 1 \end{pmatrix}.$$

*Its dual code* $\mathcal{C}^\perp$ *is the* $[n, n-1]_2$ *linear single parity-check code generated by*

$$H = \begin{pmatrix} & 1 \\ Id_{n-1} & \vdots \\ & 1 \end{pmatrix}.$$

**Proposition 2.11.** *Let* $q$ *be a prime power and* $k \leq n$ *be positive integers. Let* $\mathcal{C}$ *be an* $[n,k]_q$ *linear code. Then* $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

*Proof.* Let $G$ be a generator matrix of $\mathcal{C}$ and $H$ be a parity-check matrix of $\mathcal{C}$. Since $H$ is thus a generator matrix of $\mathcal{C}^\perp$, we get that

$$(\mathcal{C}^\perp)^\perp = \{x \in \mathbb{F}_q^n \mid \langle x, y\rangle = 0 \text{ for all } y \in \mathcal{C}^\perp\}, \quad \mathcal{C}^\perp = \{y \in \mathbb{F}_q^n \mid \langle c, y\rangle = 0 \text{ for all } c \in \mathcal{C}\}.$$

Thus any $c \in \mathcal{C}$ is also in $(\mathcal{C}^\perp)^\perp$, as $\langle c, y\rangle = 0$ for all $y \in \mathcal{C}^\perp$.
Since $\mathcal{C}$ has dimension $k$ and $(\mathcal{C}^\perp)^\perp$ has dimension $k$, we get the claim.

□

A priori, the code $\mathcal{C}$ and its dual $\mathcal{C}^\perp$ have no apparent connection. If $\mathcal{C} \subseteq \mathcal{C}^\perp$, we call $\mathcal{C}$ a *self-orthogonal* code and if $\mathcal{C} = \mathcal{C}^\perp$, we call $\mathcal{C}$ a *self-dual* code.

**Example 2.12.** *We consider again the $[n,1]_2$ repetition code $\mathcal{C}$ with generator matrix*

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

*Recall that $\mathcal{C}^\perp = ker(G^\top)$, hence to show that $\mathcal{C} \subseteq \mathcal{C}^\perp$, it is enough to show that any codeword $c \in \mathcal{C}$ is such that $cG^\top = 0$. If $n$ is even, then $GG^\top = \sum_{i=1}^{n} 1 = n$ and thus any code word $c = mG$ is such that $cG^\top = mGG^\top = 0$.*

For $x \in \mathbb{F}_q^n$ and $S \subseteq \{1, \ldots, n\}$ we denote by $x_S$ the vector consisting of the entries of $x$ indexed by $S$. While for $A \in \mathbb{F}_q^{k \times n}$, we denote by $A_S$ the matrix consisting of the columns of $A$ indexed by $S$. Similarly, we denote by $\mathcal{C}_S$ the code consisting of the codewords $c_S$, i.e.,

$$\mathcal{C}_S = \{c_s \mid c \in \mathcal{C}\}.$$

An $[n,k]_q$ linear code can be completely defined by having access only to (correctly chosen) $k$ positions. The following concept characterizes such defining sets.

**Definition 2.13** (Information Set)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n,k]_q$ linear code. Then, a set $I \subset \{1, \ldots, n\}$ of size $k$ is called an *information set* of $\mathcal{C}$ if

$$| \mathcal{C} | = | \mathcal{C}_I | .$$

**Exercise 2.14.** *How many information sets can an $[n,k]_q$ linear code have at most?*

**Proposition 2.15.** *Let $\mathcal{C}$ be an $[n,k]_q$ linear code, $I$ an information set and let $G$ be a generator matrix . The matrix $G_I$ is an invertible matrix.*

*Proof.* Since $\langle G_I \rangle = \mathcal{C}_I$ has dimension $k$, we immediately get that $G_I \in \mathbb{F}_q^{k \times k}$ has full rank. □

**Proposition 2.16.** *Let $\mathcal{C}$ be an $[n,k]_q$ linear code, $I$ an information set and $H$ a parity-check matrix. If $I^C := \{1, \ldots, n\} \setminus I$ is the complement set of $I$, then, $H_{I^C}$ is an invertible matrix.*

**Exercise 2.17.** *Prove Proposition 2.16.*

**Exercise 2.18.** *Let $\mathcal{C}$ be the code generated by $G \in \mathbb{F}_5^{2 \times 4}$, given as*

$$G = \begin{pmatrix} 1 & 3 & 2 & 3 \\ 0 & 4 & 4 & 3 \end{pmatrix}.$$

*Determine all information sets of this code.*

Since $G_I$ and $H_{I^C}$ are invertible, we can apply row operations $U$, respectively $U'$ to get

$$(UG)_I = \mathrm{Id}_k, \quad (U'H)_{I^C} = \mathrm{Id}_{n-k},$$

which leads to the following definition of a *systematic form.*

**Definition 2.19** (Systematic Form)**.** Let $k \leq n$ be positive integers and $\mathcal{C}$ be an $[n,k]_q$ linear code. Then, there exist some $n \times n$ permutation matrix $P$ and some invertible matrix $U \in \mathbb{F}_q^{k\times k}$ that bring $G$ in *systematic form*, i.e.,

$$UGP = \begin{pmatrix} \mathrm{Id}_k & A \end{pmatrix},$$

where $A \in \mathbb{F}_q^{k\times(n-k)}$. Similarly, there exist some $n \times n$ permutation matrix $P'$ and some invertible matrix $U' \in \mathbb{F}_q^{(n-k)\times(n-k)}$, that bring $H$ into systematic form, i.e.,

$$U'HP' = \begin{pmatrix} B & \mathrm{Id}_{n-k} \end{pmatrix},$$

where $B \in \mathbb{F}_q^{(n-k)\times k}$.

As we have not introduced permutation equivalence yet, we may think of the standard form as follows: Let $I \subseteq \{1,\dots,n\}$ be an information set and $G$ a generator matrix of $\mathcal{C}$, then there exists $U \in \mathrm{GL}_q(k)$ such that

$$(UG)_I = \mathrm{Id}_k, \quad \text{and} \quad (UG)_{I^C} = A,$$

for some $A \in \mathbb{F}_q^{k\times(n-k)}$. We will later see that permuting the identity matrix to be at the first $k$ coordinates, will not change the underlying structure.

Given a generator matrix, one can easily find a parity-check matrix of the code.

**Proposition 2.20.** *Let $\mathcal{C}$ be an $[n,k,]_q$ linear code and $G$ be a generator matrix. If $G = \begin{pmatrix} Id_k & A \end{pmatrix}$, for some $A \in \mathbb{F}_q^{k\times(n-k)}$, then $H = \begin{pmatrix} -A^\top & Id_{n-k} \end{pmatrix}$ is a parity-check matrix of $\mathcal{C}$.*

*Proof.* We clearly have $GH^\top = -A + A = 0$, thus $\mathcal{C} = \langle G\rangle \subseteq \ker(H^\top)$. As $H$ has rank $n-k$, the kernel of $H$ has dimension $k$, and is thus equal to $\mathcal{C}$. □

If the identity matrix is not in the first $k$ coordinates, we think of Proposition 2.20 as $G_I = \mathrm{Id}_k$ and $G_{I^C} = A$ and thus $H_{I^C} = \mathrm{Id}_{n-k}$ and $H_I = -A^T$.

**Proposition 2.21.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k]_q$ linear code. Then $I \subseteq \{1,\dots,n\}$ is an information set of $\mathcal{C}$ if and only if $I^C$ is an information set of $\mathcal{C}^\perp$.*

*Proof.* Let $G$ be a generator matrix of $\mathcal{C}$. Then by the definition of the systematic form, there exists a $U \in \mathrm{GL}_q(k)$ such that

$$(UG)_I = \mathrm{Id}_k, \quad \text{and} \quad (UG)_{I^C} = A,$$

for some $A \in \mathbb{F}_q^{k\times(n-k)}$. Without loss of generality, we may assume $I = \{1,\dots,k\}$. Thus, using Proposition 2.20 on $G' = UG$, we get that $H'_{I^C} = \mathrm{Id}_{n-k}$ and hence $I^C$ of size $n-k$ forms an information set for $\mathcal{C}^\perp$.

The other direction follows immediately from Proposition 2.11. □

**Example 2.22.** *Let us consider again the generator matrix of our toy example:*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{3\times 6},$$

*then a choice for $I$ would be $\{1,2,3\}$ and thus $A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ gives the parity-check matrix*

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

**Definition 2.23.** Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code. The *hull* of the code $\mathcal{C}$ is defined as

$$\mathcal{H}(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^{\perp}.$$

Clearly, $\mathcal{H}(\mathcal{C}) = \mathcal{H}(\mathcal{C}^{\perp})$, by Proposition 2.11.

**Exercise 2.24.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code with generator matrix $G \in \mathbb{F}_q^{k\times n}$ and parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$. Show that*

$$\mathcal{H}(\mathcal{C}) = \textit{ker}\left(\begin{pmatrix} G \\ H \end{pmatrix}^{\top}\right).$$

Note that the hull of a random code is with high probability trivial.

**Theorem 2.25.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code with generator matrix $G \in \mathbb{F}_q^{k\times n}$ and parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$. Then $\mathcal{H}(\mathcal{C}) = \{0\}$ with high probability, for $n$ growing.*

*Proof.* By Exercise 2.24 We are interested in the dimension of the kernel of the matrix $\begin{pmatrix} G \\ H \end{pmatrix}$, and due to the rank-nullity theorem in its rank. We can assume that $G, H$ are in systematic form, i.e.,

$$G = \begin{pmatrix} \mathrm{Id}_k & A \end{pmatrix}, \quad H = \begin{pmatrix} -A^{\top} & \mathrm{Id}_{n-k} \end{pmatrix}$$

and perform row operations to get

$$\begin{pmatrix} G' \\ H' \end{pmatrix}^{\top} = \begin{pmatrix} \mathrm{Id}_k & A \\ 0 & AA^{\top} + \mathrm{Id}_{n-k} \end{pmatrix}^{\top}.$$

Hence its rank is given by $k + \mathrm{rk}(AA^{\top} + \mathrm{Id}_{n-k})$. Assuming $A$ was a random matrix, we also have that $AA^{\top} + \mathrm{Id}_{n-k}$ has with high probability full rank. Thus, by rank-nullity we get

$$\dim(\mathcal{H}(\mathcal{C})) = \dim\left(\ker\left(\begin{pmatrix} G \\ H \end{pmatrix}^{\top}\right)\right) = n - \mathrm{rk}\left(\begin{pmatrix} G \\ H \end{pmatrix}^{\top}\right) = n - n = 0.$$

□

Let $\mathcal{C}$ be an $[n,k]_q$ linear code with $\langle G\rangle = \mathcal{C} = \ker(H^\top)$.

- The parameter $n$ is called the *length* of the code.
- The parameter $k$ is called the *dimension* of the code.
- The elements in the code are called *codewords*.
- The parameter $r = n - k$ is called the *redundancy*.
- The parameter $R = k/n$ is called the *rate* of the code.
- The matrix $G$ is called a *generator matrix* of the code.
- The matrix $H$ is called a *parity-check matrix* of the code.
- The vector $s = xH^\top$ is called the *syndrome* of $x$.
- The code $\mathcal{C}^\perp$ is called the *dual code* of $\mathcal{C}$.
- A set $I$ with $G_I \in \mathrm{GL}_q(k)$ is called *information set*.

## 2.2 Hamming Metric

Now that we have described the linearity of the vectors we can send, how do we decode and how much can we decode?

Recall that we have received $r = c + e \in \mathbb{F}_q^n$, where $c \in \mathcal{C}$ was sent. To decode, we have to find the sent $c$. However, $r$ could be the sum of any codeword and some other vector. In our toy example $r = (1,0,0,1,0,1,1,0,1)$ could also be written as

$$(1,1,1,1,1,1,1,1,1)+(0,1,1,0,1,0,0,1,0) \quad \text{or} \quad (1,0,0,1,0,0,1,0,0)+(0,0,0,0,0,1,0,0,1).$$

However these are less likely than

$$(1,0,1,1,0,1,1,0,1) + (0,0,1,0,0,0,0,0,0)$$

assuming that the channel only introduces a few errors.

Hence, we are looking for the closest codeword $c$. For this, we first have to define what we mean by closest, i.e., introduce a metric to $\mathbb{F}_q^n$.

**Definition 2.26.** A *distance* is a function $d : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{Q}$, such that

- it is positive definite, i.e., $d(x,y) = 0$ if and only if $x = y$ and $d(x,y) \geq 0$ for all $x, y \in \mathbb{F}_q^n$,
- it is symmetric, i.e., $d(x,y) = d(y,x)$ for all $x, y \in \mathbb{F}_q^n$,
- it satisfies the triangle inequality, i.e., $d(x,y) \leq d(x,z) + d(z,y)$ for all $x, y, z \in \mathbb{F}_q^n$.

We say that a distance is translation-invariant, if for all $x, y, z \in \mathbb{F}_q^n$ we have $d(x, y) = d(x + z, y + z)$.

A distance also gives raise to a weight, by defining $\text{wt}(x) = d(x, 0)$.

**Definition 2.27.** A *weight* is a function $\text{wt} : \mathbb{F}_q^n \to \mathbb{Q}$, such that

- it is positive definite, i.e., $\text{wt}(x) = 0$ if and only if $x = 0$ and $\text{wt}(x) \geq 0$ for all $x \in \mathbb{F}_q^n$,
- it is symmetric, i.e., $\text{wt}(x) = \text{wt}(-x)$ for all $x \in \mathbb{F}_q^n$,
- it satisfies the triangle inequality, i.e., $\text{wt}(x + y) \leq \text{wt}(x) + \text{wt}(y)$ for all $x, y \in \mathbb{F}_q^n$.

In the other direction, any weight also induces a distance by defining $d(x, y) = \text{wt}(x - y)$.

**Exercise 2.28.** *Show that if $d$ is a translation-invariant distance function $wt(x) = d(x, 0)$ is a weight function.*

**Exercise 2.29.** *Show that if wt is a weight function $d(x, y) = wt(x - y)$ is a distance function.*

As we are interested in the amount of positions which are erroneous, the most natural weight to consider is the *Hamming metric*.

**Definition 2.30** (Hamming Metric)**.** Let $n$ be a positive integer. For $x \in \mathbb{F}_q^n$, the *Hamming weight* of $x$ is given by the number of non-zero positions, i.e.,

$$\text{wt}_H(x) = \mid \{i \in \{1, \ldots, n\} \mid x_i \neq 0\} \mid .$$

For $x, y \in \mathbb{F}_q^n$, the *Hamming distance* between $x$ and $y$ is given by the number of positions in which they differ, i.e.,

$$d_H(x, y) = \mid \{i \in \{1, \ldots, n\} \mid x_i \neq y_i\} \mid .$$

Note that the Hamming distance is induced from the Hamming weight, that is $d_H(x, y) = \text{wt}_H(x - y)$.

**Exercise 2.31.** *Show that the Hamming weight is a weight function.*

Let $x \in \mathbb{F}_q^n$. We denote by $\text{supp}_H(x) = \{i \in \{1, \ldots, n\} \mid x_i \neq 0\}$ the *Hamming support* of $x$. The Hamming weight of $x$ is then clearly given as the size of the support: $\text{wt}_H(x) = |\text{supp}_H(x)|$.

Having defined a metric, one can also consider the minimum distance of a code, i.e., the smallest distance between any two distinct codewords.

**Definition 2.32** (Minimum Distance)**.** Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$. The *minimum Hamming distance* of $\mathcal{C}$ is denoted by $d_H(\mathcal{C})$ and given by

$$d_H(\mathcal{C}) = \min\{d_H(x, y) \mid x, y \in \mathcal{C},\ x \neq y\} = \min\{\text{wt}_H(c) \mid c \in \mathcal{C}, c \neq 0\}.$$

The minimum Hamming distance of a code turns out to be a very important parameter. Thus, whenever the minimum Hamming distance $d = d_H(\mathcal{C})$ is known, we say $\mathcal{C}$ is an $[n, k, d]_q$ linear code.

**Example 2.33.** *The $[n, 1]_2$ repetition code has minimum distance $d_H(\mathcal{C}) = n$, whereas the $[n, n-1]_2$ single parity-check code has minimum distance $d_H(\mathcal{C}^\perp) = 2$.*

When defining balls in a certain metric, we have to provide the radius and the center, e.g. we may define the Hamming ball of radius $r$ and center $x$ as

$$B_H(r, n, q, x) = \{y \in \mathbb{F}_q^n \mid d_H(x, y) \leq r\}.$$

However, to determine the size of such balls, we observe that the Hamming metric is translation invariant.

**Proposition 2.34.** *Let $q$ be a prime power and $r \leq n$ be positive integers. For any $x, x' \in \mathbb{F}_q^n$ we have*

$$|B_H(r, n, q, x)| = |B_H(r, n, q, x')| = |\{y \in \mathbb{F}_q^n \mid wt_H(y) \leq r\}| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

**Exercise 2.35.** *Prove Proposition 2.34*

The minimum distance of a code is an important parameter, since it is connected to the error correction capability of the code.

## 2.3 Error-Correction Capability

We denote by $d_H(x, \mathcal{C})$ the minimal distance between $x \in \mathbb{F}_q^n$ and a codeword in $\mathcal{C}$.

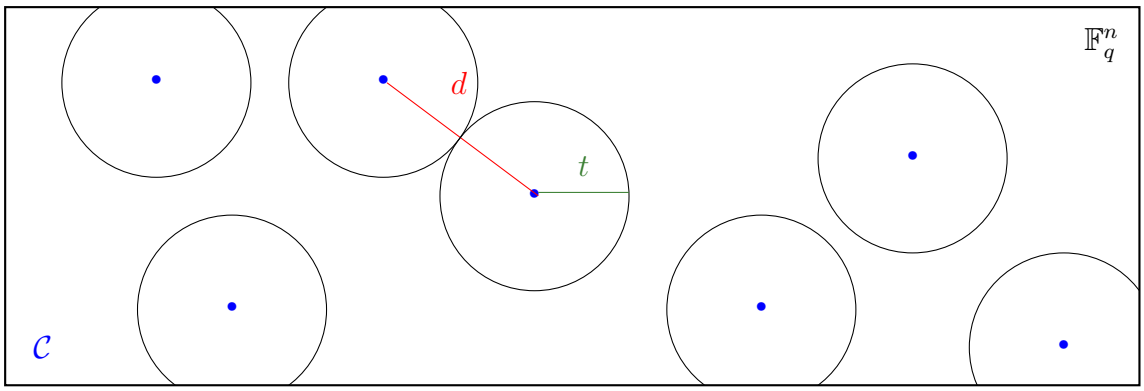
We say that a code can *correct* up to $t$ errors, if for all $r \in \mathbb{F}_q^n$ with $d_H(r, \mathcal{C}) \leq t$, there exists at most one $c \in \mathcal{C}$, such that $d_H(r, c) \leq t$. The parameter $t$ is then called the *error correction capability* of the code. Equivalently, $\mathcal{C}$ can correct $t$ errors, if the balls of radius $t$ around any distinct codewords $c \neq c'$ are disjoint: $B_H(t, n, q, c) \cap B_H(t, n, q, c') = \emptyset$.

On the other hand, we say that a code can *detect* $w$ errors, if for any two distinct codewords $c \neq c' \in \mathcal{C}$ we have $d_H(c, c') > w$. Equivalently, we may say $\mathcal{C}$ can detect $w$ errors, if the ball of radius $w$ around any codeword $c$ does not contain any other codeword: $B_H(w, n, q, c) \cap \mathcal{C} = \{c\}$. The parameter $w$ is then called the *error detection capability* of the code.

How are $d_H(\mathcal{C})$ and $t$, the error correction capability of $\mathcal{C}$, related?

If we are given the code $\mathcal{C} \subseteq \mathbb{F}_q^n$ and depict its codewords as points, then the shortest distance between two of them is given by $d_H(\mathcal{C}) = d$. To find the error correction capability, we want to draw balls around the codewords, with radius as large as possible, but such that the balls do not intersect. This results in the radius

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$




**Theorem 2.36.** *Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. Then $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ is the error correction capability of the code.*

*Proof.* Let $r \in \mathbb{F}_q^n$ be such that there exists a codeword $c \in \mathcal{C}$ with $d_H(c,r) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. Assume by contradiction, that there exists another codeword $c' \in \mathcal{C}$ with $c \neq c'$ and $d_H(c',r) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. Then, by the triangle inequality, we have that

$$d_H(c,c') \leq d_H(c,r) + d_H(c',r) \leq \left\lfloor \frac{d-1}{2} \right\rfloor + \left\lfloor \frac{d-1}{2} \right\rfloor < d,$$

contradicting that $d$ is the minimum Hamming distance of $\mathcal{C}$.

This is also the maximal value. In fact, let $c, c' \in \mathcal{C}$ be such that $d_H(c,c') = d$. Then for $t \geq \left\lfloor \frac{d-1}{2} \right\rfloor + 1$ there exist $r \in \mathbb{F}_q^n$ with $d_H(c,r) \leq t$ and $d_H(c',r) \leq t$. Without loss of generality, we may assume that $c, c'$ differ in the first $d$ positions, i.e., $c = (c_1, \ldots, c_d, x_1, \ldots, x_{n-d})$ and $c' = (c'_1, \ldots, c'_d, x_1, \ldots, x_{n-d})$. Then, we can set $r = (c_1, \ldots, c_t, c'_{t+1}, \ldots, c'_d, x_1, \ldots, x_{n-d})$ and check $d_H(c,r) \leq d - t \leq t$ and $d_H(c',r) \leq t$. □

**Exercise 2.37.** *Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. Prove that the error detection capability is given by $w = d - 1$.*

*Decoding* denotes a function, that takes as input $r = c + e \in \mathbb{F}_q^n$ and returns the closest codeword, $c \in \mathcal{C}$, such that $d_H(r,c) \leq t$.

$$\text{Dec} : \mathbb{F}_q^n \to \mathcal{C}, \quad r \mapsto c = \text{argmin}\{d_H(r,x) \mid x \in \mathcal{C}\}.$$

Where "function" is bad word, as there might be $r \in \mathbb{F}_q^n$ which are not decodable. It should rather be thought as

$$\text{Dec} : \bigcup_{c \in \mathcal{C}} B_H(t,n,q,c) \to \mathcal{C}, \quad r \mapsto c = \text{argmin}\{d_H(r,x) \mid x \in \mathcal{C}\}.$$

The most interesting codes for applications are codes with an efficient decoding algorithm, i.e., where we can compute the function Dec efficiently.

How can we determine the minimum distance of a code?

**Theorem 2.38.** *Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n,k]_q$ linear code. Let $H$ be a parity-check matrix of $\mathcal{C}$. Then, $\mathcal{C}$ has minimum distance $d$ if and only if every $d-1$ columns of $H$ are linearly independent and there exist $d$ columns, which are linearly dependent.*

*Proof.* For the first direction, let $c \in \mathcal{C}$ be the minimal weight codeword, i.e., $\mathrm{wt}_H(c) = d$. Since $c \in \ker(H^\top)$, we get $cH^\top = 0$ and thus the $d$ columns $h_i$ of $H$, indexed by $i \in \mathrm{supp}_H(c)$ are linearly dependent as $\sum_{i \in \mathrm{supp}_H(c)} c_i h_i = 0$. On the other hand, any $c \in \mathcal{C}$ with weight $< d$ must be the zero codeword. Thus, if a linear combination of less than $d$ columns of $H$ gives zero, the scalars are zero, i.e., any $d-1$ columns are linearly independent.

For the other direction, assume there exist $d$ linearly dependent columns $h_i$ of $H$ for $i \in I$ and $|I| = d$, that is there exist $\lambda_i \in \mathbb{F}_q^\star$ such that $\sum_{i \in I} \lambda_i h_i = 0$. Then $c$ with support $I$ and entries $c_i = \lambda_i$ for $i \in I$ is such that $\mathrm{wt}_H(c) = d$ and $cH^\top = 0$, thus $c \in \mathcal{C}$ and hence $d_H(\mathcal{C}) \leq d$. Since any set of $d-1$ columns are linearly dependent, by the same argument, there is no non-zero codeword with weight less than $d$, thus $d_H(\mathcal{C}) \geq d$.

□

**Example 2.39.** *Let us consider again the binary code $\mathcal{C}$ from the toy example, with parity-check matrix*

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

*We can find 3 columns which are linearly dependent, for example the columns indexed by $\{1,4,6\}$, as*

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

*And any two columns are linearly independent, thus by Theorem 2.38, we get $d_H(\mathcal{C}) = 3$. Thus, we can correct $t = 1$ error and detect 2 errors.*

**Theorem 2.40.** *Let $q$ be a prime power and $k, d \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. Then every subset of $\{1,\ldots,n\}$ of size $n-d+1$ contains an information set. Furthermore, $d$ is the largest number with this property.*

*Proof.* Let $G$ be a generator matrix of $\mathcal{C}$ and consider any set $X \subset \{1,\ldots,n\}$ of size $s$. Without loss of generality, we may assume $X = \{1,\ldots,s\}$ and split the generator matrix as

$$G = \begin{pmatrix} A & B \end{pmatrix},$$

where $A \in \mathbb{F}_q^{k \times s}$, $B \in \mathbb{F}_q^{k \times (n-s)}$. Assume that $X$ does not contain an information set, thus $\mathrm{rk}(A) < k$. Thus, there exists a non-trivial linear combination of the rows of $A$ which gives 0 and hence there exists a codeword $c$ having in the first $s$ positions 0's. Since the rows of $G$ are linearly independent, $c \neq 0$ and hence $d \leq \mathrm{wt}_H(c) \leq n-s$.

Thus, whenever $s > n-d$, e.g. for $s = n-d+1$ the set $X$ must contain an information set. □

Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code with $\langle G\rangle = \mathcal{C} = \ker(H^\top)$.

- The parameter $n$ is called the *length* of the code.
- The parameter $k$ is called the *dimension* of the code.
- The elements in the code are called *codewords*.
- The parameter $r = n - k$ is called the *redundancy*.
- The parameter $R = k/n$ is called the *rate* of the code.
- The matrix $G$ is called a *generator matrix* of the code.
- The matrix $H$ is called a *parity-check matrix* of the code.
- The vector $s = xH^\top$ is called the *syndrome* of $x$.
- The code $\mathcal{C}^\perp$ is called the *dual code* of $\mathcal{C}$.
- A set $I$ with $G_I \in \mathrm{GL}_q(k)$ is called *information set*.
- The parameter $d$ is called the *minimum Hamming distance* of the code.
- The parameter $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ is called the *error correction capability* of the code.

## 2.4 Subcodes and Supercodes

If a code contains a smaller code, we call this smaller code a *subcode* and the larger code a *supercode*:

**Definition 2.41.** Let $q$ be a prime power and $k' \leq k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code and $\mathcal{C}'$ be a $[n,k']_q$ linear code. If $\mathcal{C}' \subseteq \mathcal{C}$, then $\mathcal{C}'$ is a *subcode* of $\mathcal{C}$ and $\mathcal{C}$ is a *supercode* of $\mathcal{C}'$.

Given a generator matrix $G \in \mathbb{F}_q^{k\times n}$ of a $[n,k]_q$ code, then deleting $\ell$ rows will give a generator matrix $G' \in \mathbb{F}_q^{k'\times n}$ of a $[n,k']_q$ subcode $\mathcal{C}'$, where $k' = k - \ell$.

Even more is true:

**Proposition 2.42.** *Let $q$ be a prime power and $k' \leq k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code and $\mathcal{C}'$ be a $[n,k']_q$ linear code with $\mathcal{C}' \subseteq \mathcal{C}$. Then for any generator matrix $G \in \mathbb{F}_q^{k\times n}$ of $\mathcal{C}$ there exist some $S \in \mathrm{GL}_q(k)$ such that the first $k'$ rows of $SG$ generate $\mathcal{C}'$.*

**Exercise 2.43.** *Prove Proposition 2.42.*

For the parity-check matrix the opposite is happening. In fact, while for the code $\mathcal{C}$, we have that the subcode $\mathcal{C}'$ is smaller and contained in $\mathcal{C}$, their dual codes are such that

$$\mathcal{C}^\perp \subseteq \mathcal{C}'^\perp.$$

**Proposition 2.44.** *Let $q$ be a prime power and $k' \leq k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code and $\mathcal{C}'$ be a $[n,k']_q$ linear code with $\mathcal{C}' \subseteq \mathcal{C}$. Then*

$$\mathcal{C}^\perp \subseteq \mathcal{C}'^\perp.$$

*Proof.* Let us consider the generator matrices $G \in \mathbb{F}_q^{k\times n}$ and $G' \in \mathbb{F}_q^{k'\times n}$ of $\mathcal{C}$, respectively $\mathcal{C}'$ and the parity-check matrices $H \in \mathbb{F}_q^{(n-k)\times n}$ and $H' \in \mathbb{F}_q^{(n-k')\times n}$. Recall that $GH^\top = 0$ and by Proposition 2.42 we may assume that $G = \begin{pmatrix} G' \\ X \end{pmatrix}$, where $X \in \mathbb{F}_q^{\ell\times n}$ for $\ell = k - k'$. Since then

$$GH^\top = \begin{pmatrix} G' \\ X \end{pmatrix} H^\top = 0$$

we also get that $G'H^\top = 0$ and hence $\mathrm{Im}(H) = \mathcal{C}^\perp \subseteq \mathcal{C}'^\perp$.

□

Hence, we can apply the same argument as in Proposition 2.42 to their duals to get

**Corollary 2.45.** *Let $q$ be a prime power and $k' \leq k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code and $\mathcal{C}'$ be a $[n,k']_q$ linear code with $\mathcal{C}' \subseteq \mathcal{C}$. Then for any parity-check matrix $H' \in \mathbb{F}_q^{(n-k')\times n}$ of $\mathcal{C}'$ there exist some $S \in GL_q(n-k')$ such that the first $n-k$ rows of $SH'$ are a parity-check matrix for $\mathcal{C}$.*

**Exercise 2.46.** *Let $q$ be a prime power and $k' \leq k \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k]_q$ linear code and $\mathcal{C}'$ be a $[n,k']_q$ linear code with $\mathcal{C}' \subseteq \mathcal{C}$. Show that $d_H(\mathcal{C}) \leq d_H(\mathcal{C}')$.*

**Example 2.47.** *Any code of dimension $1 \leq k < n$ is a subcode of the $[n,n]_q$ code $\mathbb{F}_q^n$ and a supercode of the $[n,0]_q$ code $\{0\}$.*

*Let us consider again the generator matrix of our toy example:*

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{3\times 6},$$

*with the parity-check matrix*

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{3\times 6}.$$

*Then a subcode of $\langle G\rangle$ is generated by*

$$G' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{2\times 6}$$

*and a parity-check matrix is then given by*

$$H' = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{F}_2^{4\times 6}.$$

## 2.5 Counting Codes

If we fix $n, k$ and $q$, how many $[n, k]_q$ linear codes are there?

**Definition 2.48.** The *Gaussian coefficient* or the $q$-binomial coefficient is defined as

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \prod_{i=0}^{k-1} \frac{q^n - q^i}{q^k - q^i} = \frac{\prod_{i=1}^{n}(1-q^i)}{\prod_{i=1}^{k}(1-q^i)\prod_{i=1}^{n-k}(1-q^i)}.$$

To handle the Gaussian coefficient, we often use the following bounds and limits:

**Proposition 2.49.** *Let $q$ be a prime power $k \leq n$ be positive integers.*

- *The following identity holds*
$$\begin{bmatrix} n \\ k \end{bmatrix}_q = q^{(n-k)k} \begin{bmatrix} n \\ k \end{bmatrix}_{1/q}.$$
- *If $k$ is a constant in $n$, then*
$$\lim_{n\to\infty} \begin{bmatrix} n \\ k \end{bmatrix}_{1/q} = \prod_{i=1}^{k}(1-q^{-i})^{-1}.$$
- *If $k = Rn$ for $0 < R < 1$, then*
$$\lim_{n\to\infty} \begin{bmatrix} n \\ Rn \end{bmatrix}_{1/q} = \prod_{i=1}^{\infty}(1-q^{-i})^{-1},$$
*whereas for $R \in \{0, 1\}$ we have*
$$\begin{bmatrix} n \\ Rn \end{bmatrix}_{1/q} = 1.$$
- *We can bound the Gaussian binomial as*
$$q^{(n-k)k} \leq \begin{bmatrix} n \\ k \end{bmatrix}_q \leq q^{(n-k)k} \prod_{i=1}^{\infty}(1-q^{-i})^{-1}.$$
- *Since*
$$\lim_{q\to\infty} \prod_{i=1}^{\infty}(1-q^{-i})^{-1} = 1,$$
*we have that*
$$\begin{bmatrix} n \\ k \end{bmatrix}_q \sim q^{(n-k)k},$$
*for $q \to \infty$.*

- *We can also bound $\prod_{i=1}^{\infty}(1-q^{-i})^{-1}$, which is called the Euler function, usually denoted by $(1/q)_\infty$, by*

$$\frac{1}{4} \leq \prod_{i=1}^{\infty}(1-q^{-i})^{-1} \leq 1.$$

**Proposition 2.50.** *Let $q$ be a prime power and $k \leq n$ be positive integers. The number of $[n,k]_q$ linear codes is given by $\begin{bmatrix} n \\ k \end{bmatrix}_q$.*

*Proof.* To determine a $k$-dimensional subspace in $\mathbb{F}_q^n$, we need to choose $k$ linearly independent vectors from $\mathbb{F}_q^n$, which act as a basis of $\mathcal{C}$. The first vector $b_1$ should be non-zero, thus we have $q^n - 1$ choices. The second vector $b_2$ must be chosen outside the span of the first vector $b_1$. Since, the first vector generates a subspace of dimension 1, we have $q^n - q$ choices. We continue in this way, i.e., for the $i$th vector, we require $b_i \in \mathbb{F}_q^n \setminus \langle b_1, \ldots, b_{i-1} \rangle$, which gives $q^n - q^{i-1}$ choices. Thus, there are $\prod_{i=0}^{k-1}(q^n - q^i)$ choices for linearly independent vectors, spanning a subspace of dimension $k$.

However, some of these subspaces might be the same. Thus, we need to divide this number by the number of $k$ linearly independent vectors, which span the same space. This follows a similar argument: having fixed $\mathcal{C}$ of dimension $k$, how many bases does $\mathcal{C}$ have, is equivalent to asking for the number of $k$ linearly independent vectors in $\mathcal{C} \cong \mathbb{F}_q^k$. Thus, by setting $n = k$ in the previous counting argument, we get $\prod_{i=0}^{k-1}(q^k - q^i)$ and the final formula

$$\frac{\prod_{i=0}^{k-1}(q^n - q^i)}{\prod_{i=0}^{k-1}(q^k - q^i)} = \begin{bmatrix} n \\ k \end{bmatrix}_q.$$

□

**Exercise 2.51.** *Given $\mathcal{C}$ a $[n,k]_q$ linear code. How many subcodes does $\mathcal{C}$ have?*

Can we choose any $n, k, d$?

On one hand, we want to be able to encode many messages, thus we would like to have a large $k$. At same time want to be able to correct many errors, which requires a large $d$. For fixed $n$, having both parameters $k, d$ large is not possible. Thus, we are interested in providing bounds on these parameters and finding optimal codes, e.g. codes which can correct the most errors for given $n, k$.

# 3 MDS Codes

As we have seen, the minimum distance $d$ of a code is an important parameter, which determines how many errors the code can correct. Clearly, we cannot have a large amount of codewords $q^k$ while maintaining that they are far apart, i.e., of distance at least $d$.

Thus, a large branch in coding theory is considering bounds on the dimension $k$ when the length $n$ and the minimum distance $d$ are fixed, or equivalently, bounds on the minimum distance $d$ when the length $n$ and the dimension $k$ are fixed. The most prominent such upper bound on $k$ (respectively on $d$) is the *Singleton bound.*

## 3.1 Singleton Bound

One of the most important bounds in coding theory is the Singleton bound, which provides an upper bound on the minimum distance of a code.

**Theorem 3.1** (Singleton Bound)**.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k]_q$ linear code. Then,*

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

There exist many different proofs for this bound, using concepts we have not introduced yet. However, the easiest one follows directly form the systematic form of a generator matrix.

*Proof.* Let $G \in \mathbb{F}_q^{k\times n}$ be a generator matrix of $\mathcal{C}$ and let $I \subset \{1,\dots,n\}$ be an information set. Recall that there exist some $U \in \mathrm{GL}_k(q)$, such that we may write $(UG)_I = \mathrm{Id}_k$ and $(UG)_{I^C} = A$ for some $A \in \mathbb{F}_q^{k\times(n-k)}$. As any row $g$ of $UG$ is again a codeword in $\mathcal{C}$, we have by definition $d_H(\mathcal{C}) \leq \mathrm{wt}_H(g)$. By the form of $g$ we see that $\mathrm{wt}_H(g_I) = 1$ and as $g_{I^C} \in \mathbb{F}_q^{n-k}$ we get $\mathrm{wt}_H(g_{I^C}) \leq n-k$. Thus,

$$d_H(\mathcal{C}) \leq \mathrm{wt}_H(g) \leq 1 + n - k.$$

□

A code that achieves the Singleton bound is called a *maximum distance separable* (MDS) code. MDS codes are of immense interest, since they can correct the maximal amount of errors for fixed code parameters $n, k$. In fact, their error correction capability is given by

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor.$$

**Theorem 3.2.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k]_q$ linear code. Then, the following are equivalent:*

1. *$\mathcal{C}$ is an MDS code.*
2. *Every subset of $\{1,\dots,n\}$ of size $k$ is an information set.*

*3. $\mathcal{C}^\perp$ is an MDS code.*

*Proof.* We start by showing 1) $\implies$ 2). From Theorem 2.40, we know that any subset of $\{1, \dots, n\}$ of size $n-d+1 = k$ contains an information set of size $k$, i.e., any subset of $\{1, \dots, n\}$ of size $k$ is an information set.

This also gives the other direction 2) $\implies$ 1) as Theorem 2.40 also states that $d$ is the largest number with this property, i.e., since every subset of size $k$ is an information set, $k = n - d + 1$ and thus $\mathcal{C}$ is MDS.

Note that this argument can also be applied to the dual code: if $\mathcal{C}^\perp$ is MDS, then any subset of $\{1, \dots, n\}$ of size $n - k$ is an information set for $\mathcal{C}^\perp$. Recall from Proposition 2.21 that $J$ is an information set of $\mathcal{C}^\perp$ if and only if $J^C$ is an information set of $\mathcal{C}$, proving that 3) $\Leftrightarrow$ 2). □

## 3.2 Trivial MDS Codes

Some examples of MDS codes are quite trivial.

**Example 3.3.** *Let $\mathcal{C} = \mathbb{F}_q^n$ be the $[n, n, 1]_q$ linear code. Then as*

$$n - n + 1 = 1 = d_H(\mathcal{C})$$

*we get that $\mathcal{C}$ is MDS. Thus, by Theorem 3.2, the dual code is also MDS. Since $\mathcal{C}^\perp = \{0\}$ is the $[n, 0]_q$ linear code, we see that we should define*

$$d_H(\{0\}) = n + 1.$$

**Example 3.4.** *Let us consider a code $\mathcal{C}$ with dimension 1. If the code is non-degenerate, then $d_H(\mathcal{C}) = n$. Thus, any non-degenerate code of dimension 1 is an MDS code, as*

$$d_H(\mathcal{C}) = n = n - 1 + 1.$$

*Thus, again by Theorem 3.2, we get that the dual code $\mathcal{C}^\perp$ of dimension $n - 1$ is also MDS. Thus,*

$$d_H(\mathcal{C}^\perp) = n - (n - 1) + 1 = 2.$$

*In fact, $\mathcal{C}^\perp$ does not contain any vector of Hamming weight 1. If a generator matrix of $\mathcal{C}$ is given by $G = [g_1, \dots, g_n]$ with $g_i \in \mathbb{F}_q^\star$, then $G$ acts as parity-check matrix for $\mathcal{C}^\perp$ and for any vector $x$ with $\text{supp}_H(x) = \{i\}$, we get $Gx^\top = x_i g_i \neq 0$.*

*On the other hand, we have, for example $x = (g_2, -g_1, 0 \dots, 0) \in \mathcal{C}^\perp$.*

Apart from these trivial codes, we also have a large family of codes, for any $k \leq n \leq q$, which are MDS, called Reed-Solomon codes.

## 3.3 Reed-Solomon Codes

The most famous example is the family of Reed-Solomon (RS) codes.

**Definition 3.5.** Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$ are pairwise distinct. The Reed-Solomon code $\mathcal{RS}_{q,n,k}(\alpha)$ of dimension $k$ is defined as

$$\mathcal{RS}_{q,n,k}(\alpha) = \{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f(x) \in \mathbb{F}_q[x], \deg(f) < k\}.$$

Thus, any codeword in $\mathcal{RS}_{q,n,k}(\alpha)$ is an evaluation of a polynomial of degree up to $k$, in the evaluation points $\alpha_1, \ldots, \alpha_n$.

Why is this a linear code, though?

Let us show that the Vandermonde matrix is a generator matrix of $\mathcal{RS}_{q,n,k}(\alpha)$. The Vandermonde matrix $V_{q,n,k}(\alpha)$ is given by

$$V_{q,n,k}(\alpha) = \begin{pmatrix} 1 & \ldots & 1 \\ \alpha_1 & \ldots & \alpha_n \\ \vdots & & \vdots \\ \alpha_1^{k-1} & \ldots & \alpha_n^{k-1} \end{pmatrix} \in \mathbb{F}_q^{k \times n}.$$

(Note that the usual definition of a Vandermonde matrix is the transpose of this matrix).

**Exercise 3.6.** *Show that*

$$V = \begin{pmatrix} 1 & \ldots & 1 \\ \alpha_1 & \ldots & \alpha_k \\ \vdots & & \vdots \\ \alpha_1^{k-1} & \ldots & \alpha_k^{k-1} \end{pmatrix}$$

*has determinant*

$$\mathit{det}(V) = \prod_{0 \leq i < j \leq k} (\alpha_j - \alpha_i)$$

*and thus, every maximal minor of $V_{q,n,k}(\alpha)$ is non-zero.*

**Proposition 3.7.** *Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$ are pairwise distinct. Then $\langle V_{q,n,k}(\alpha) \rangle = \mathcal{RS}_{q,n,k}(\alpha)$.*

*Proof.* Any codeword of $\mathcal{RS}_{q,n,k}(\alpha)$ is of the form

$$c = (f(\alpha_1), \ldots, f(\alpha_n)) = \left( \sum_{i=0}^{k-1} f_i \alpha_1^i, \ldots, \sum_{i=0}^{k-1} f_i \alpha_n^i \right),$$

for $f(x) = \sum_{i=0}^{k-1} f_i x^i$.

Thus,

$$c = (f_0, \ldots, f_{k-1}) V_{q,n,k}(\alpha)$$

and in turn $\mathcal{RS}_{q,n,k}(\alpha) \subseteq \langle V_{q,n,k}((\alpha)\rangle$.

We note that $|\mathcal{RS}_{q,n,k}(\alpha)| = q^k$, as there are $q^k$ many polynomials of degree up to $k$ over $\mathbb{F}_q$. Since also $V_{q,n,k}(\alpha)$ has rank $k$, we get that $\langle V_{q,n,k}(\alpha)\rangle = \mathcal{RS}_{q,n,k}(\alpha)$.

□

**Proposition 3.8.** *Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$'s are pairwise distinct. The Reed-Solomon code $\mathcal{RS}_{q,n,k}(\alpha)$ is an MDS code.*

*Proof.* Since $\mathcal{RS}_{q,n,k}(\alpha)$ is a $[n,k]_q$-linear code, we must have $d_H(\mathcal{RS}_{q,n,k}(\alpha)) \leq n - k + 1$. For the other direction, i.e., $d_H(\mathcal{RS}_{q,n,k}(\alpha)) \geq n - k + 1$, let us consider a codeword $c = (f(\alpha_1), \ldots, f(\alpha_n))$, where $\deg(f) = k - 1$.

If, by contradiction, any non-zero codeword has weight smaller than $n - k + 1$, i.e.,

$$\mathrm{wt}_H(c) = n - \ell < n - k + 1$$

then there exist $\ell$ evaluation points $\alpha_{i_1}, \ldots, \alpha_{i_\ell} \in \{\alpha_1, \ldots, \alpha_n\}$ with

$$f(\alpha_{i_1}) = \cdots = f(\alpha_{i_\ell}) = 0.$$

Since $\deg(f) = m \leq k - 1$, $f$ can have at most $m \leq k - 1$ roots in $\mathbb{F}_q$, however,

$$\ell > n - (n - k + 1) = k - 1 \geq m.$$

Thus, $f(x) = 0$ and $c = 0$ leading to the desired contradiction. □

**Example 3.9.** *Let us consider $\alpha = (1, 2, 4, 3)$ and $\mathcal{RS}_{5,4,3}(\alpha) \subset \mathbb{F}_5^4$, which is generated by*

$$V_{5,4,3}(\alpha) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \end{pmatrix}.$$

*On the other hand, $\mathcal{RS}_{5,4,2}(\alpha) \subset \mathcal{RS}_{5,4,3}(\alpha)$, which is generated by*

$$V_{5,4,2}(\alpha) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \end{pmatrix}.$$

We can easily see that

$$\mathcal{RS}_{q,n,\ell}(\alpha) \subseteq \mathcal{RS}_{q,n,k}(\alpha)$$

is a subcode for $\ell \leq k$. The dual code of a Reed-Solomon (RS) code is not necessarily a RS code. To see this, let us first introduce the systematic form of a generator matrix of a RS code, to find the parity-check matrix.

**Proposition 3.10.** *Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$'s are pairwise distinct. Then $\mathcal{RS}_{q,n,k}(\alpha) = \langle G \rangle$, where*

$$G = \begin{pmatrix} Id_k & C \end{pmatrix}$$

*and*

$$H = \begin{pmatrix} -C^\top & Id_{n-k}, \end{pmatrix}$$

*where $C \in \mathbb{F}_q^{k \times (n-k)}$ is a Cauchy matrix of the form $c_{ij} = \frac{a_i b_j}{x_i + y_j}$, with*

$$\begin{aligned} x_i &= -\alpha_i, & y_j &= \alpha_{j+k}, \\ a_i &= \frac{1}{\prod_{t=1, t\neq i}^{k} (\alpha_i - \alpha_t)}, & b_j &= \prod_{t=1}^{k} (\alpha_{j+k} - \alpha_t). \end{aligned}$$

We omit the proof, but it essentially computes the inverse of the first $k \times k$ Vandermonde matrix, $V_{q,n,k}(\alpha_1, \ldots, \alpha_k)^{-1}$, and multiplies it to the remaining $k \times (n-k)$ Vandermonde matrix, i..e, $V_{q,k,k}(\alpha_1, \ldots, \alpha_k)^{-1} V_{q,n-k,k}(\alpha_{k+1}, \ldots, \alpha_n)$.

**Example 3.11.** *Let us consider again $\alpha = (1, 2, 4, 3)$ and $\mathcal{RS}_{5,4,3}(\alpha) \subset \mathbb{F}_5^4$, then in systematic form, we have a generator matrix*

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

*Hence*

$$H = \begin{pmatrix} 2 & 4 & 3 & 1 \end{pmatrix},$$

*which clearly does not generate a RS code.*

However, it is almost a RS code up to scaling of the columns. Thus, let us introduce generalized Reed-Solomon codes.

## 3.4 Generalized Reed-Solomon Codes

A natural generalization of the family of Reed-Solomon codes, is to weight each entry $i$ with some scalar $\beta_i \neq 0$, giving rise to the Generalized Reed-Solomon (GRS) codes.

**Definition 3.12.** Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ where the $\alpha_i$'s are pairwise distinct and $\beta = (\beta_1, \ldots, \beta_n) \in (\mathbb{F}_q^\star)^n$. The *Generalized Reed-Solomon code* $\mathcal{GRS}_{q,n,k}(\alpha, \beta)$ of dimension $k$ is defined as

$$\mathcal{GRS}_{q,n,k}(\alpha, \beta) = \{(\beta_1 f(\alpha_1), \ldots, \beta_n f(\alpha_n)) \mid f(x) \in \mathbb{F}_q[x], \deg(f) < k\}.$$

We first note that a GRS code is generated by a weighted Vandermonde matrix, that is

$$V_{q,n,k}(\alpha,\beta) = \begin{pmatrix} \beta_1 & \dots & \beta_n \\ \beta_1\alpha_1 & \dots & \beta_n\alpha_n \\ \vdots & & \vdots \\ \beta_1\alpha_1^{k-1} & \dots & \beta_n\alpha_n^{k-1} \end{pmatrix} \in \mathbb{F}_q^{k\times n}.$$

**Proposition 3.13.** *Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1,\dots,\alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$ are pairwise distinct and $\beta = (\beta_1,\dots,\beta_n) \in (\mathbb{F}_q^\star)^n$. Then $\langle V_{q,n,k}(\alpha,\beta)\rangle = \mathcal{GRS}_{q,n,k}(\alpha,\beta)$.*

**Exercise 3.14.** *Prove Proposition 3.13.*

Note that GRS codes are still MDS codes.

**Proposition 3.15.** *Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1,\dots,\alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$ are pairwise distinct and $\beta = (\beta_1,\dots,\beta_n) \in (\mathbb{F}_q^\star)^n$. Then $\mathcal{GRS}_{q,n,k}(\alpha,\beta)$ is an MDS code.*

**Exercise 3.16.** *Prove Proposition 3.15.*

**Proposition 3.17.** *Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1,\dots,\alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$ are pairwise distinct and $\beta = (\beta_1,\dots,\beta_n) \in (\mathbb{F}_q^\star)^n$. Then $\mathcal{GRS}_{q,n,k}^\perp(\alpha,\beta) = \mathcal{GRS}_{q,n,n-k}(\alpha,\gamma)$, where*

$$\gamma_i = \beta_i^{-1} \prod_{j=1,j\neq i}^{n} (\alpha_i - \alpha_j)^{-1}$$

*for all $i \in \{1,\dots,n\}$.*

*Proof.* Let us define

$$L_i(x) = \prod_{j=1,j\neq i}^{n} (x - \alpha_j)$$

and $\gamma_i = \beta_i^{-1} L_i(\alpha_i)^{-1}$ for all $i \in \{1,\dots,n\}$. Let $c \in \mathcal{GRS}_{q,n,k}(\alpha,\beta)$ and $c' \in \mathcal{GRS}_{q,n,n-k}(\alpha,\gamma)$ be arbitrary codewords. We want to show that $\langle c, c'\rangle = 0$.

We can define $f(x), f'(x) \in \mathbb{F}_q[x]$ of degree $\deg(f) < k, \deg(f') < n-k$, such that

$$c = (\beta_1 f(\alpha_1),\dots,\beta_n f(\alpha_n)),$$
$$c' = (\gamma_1 f'(\alpha_1),\dots,\gamma_n f'(\alpha_n)).$$

Clearly, $\deg(f\cdot f') < n-1$, as the degree is at most $k-1+(n-k-1) = n-2$.

Recall that by Lagrange interpolation for any $g(x) \in \mathbb{F}_q[x]$, of degree $\leq n-1$, we have that

$$g(x) = \sum_{i=1}^{n} \frac{L_i(x)}{L_i(\alpha_i)} g(\alpha_i).$$

Thus, applying this to $g = f \cdot f'$ we get

$$(f \cdot f')(x) = \sum_{i=1}^{n} \frac{L_i(x)}{L_i(\alpha_i)} f(\alpha_i) f'(\alpha_i).$$

As the degree of $f \cdot f'$ is strictly less than $n-1$, we have that the coefficient of $x^{n-1}$ of $(f \cdot f')(x)$ is 0. On the other hand, the coefficient of $x^{n-1}$ of $L_i(x)$ is 1, and hence

$$\begin{aligned} 0 &= \sum_{i=1}^{n} \frac{1}{L_i(\alpha_i)} f(\alpha_i) f'(\alpha_i) \\ &= \sum_{i=1}^{n} (\beta_i f(\alpha_i)) \left( \frac{\beta_i^{-1}}{L_i(\alpha_i)} f'(\alpha_i) \right) \\ &= \sum_{i=1}^{n} (\beta_i f(\alpha_i))(\gamma_i f'(\alpha_i)) \\ &= \sum_{i=1}^{n} c_i c_i' = \langle c, c' \rangle. \end{aligned}$$

□

**Example 3.18.** *Let us consider $\mathbb{F}_7$ and $\alpha = (2,5,4), \beta = (1,2,3)$, then*

$$V_{5,3,2}(\alpha, \beta) = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 5 \end{pmatrix}$$

*and $\mathcal{GRS}_{5,3,2}(\alpha, \beta) = \langle V_{5,3,2}(\alpha, \beta) \rangle$ while $\gamma = (6,5,3)$ and*

$$V_{5,3,1}(\alpha, \gamma) = \begin{pmatrix} 6 & 5 & 3 \end{pmatrix}$$

*and $\mathcal{GRS}_{5,3,2}^{\perp}(\alpha, \beta) = \langle V_{5,3,1}(\alpha, \gamma) \rangle$.*

**Exercise 3.19.** *Show that the examples of MDS codes from before, i.e., $\mathcal{C}$ of dimension in $\{n, 0, 1, n-1\}$ are all GRS codes if $n \leq q$.*

We can also see now, that the dual of a RS code is a GRS code.

**Corollary 3.20.** *Let $q$ be a prime power and $k \leq n \leq q$ be positive integers. Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$, where the $\alpha_i$ are pairwise distinct. Then $\mathcal{RS}_{q,n,k}^{\perp}(\alpha) = \mathcal{GRS}_{q,n,n-k}(\alpha, \gamma)$, where*

$$\gamma_i = \prod_{j=1, j \neq i}^{n} (\alpha_i - \alpha_j)^{-1}$$

*for all $i \in \{1, \ldots, n\}$.*

**Exercise 3.21.** *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code. Show that $d_H(\mathcal{C}) + d_H(\mathcal{C}^{\perp}) \leq n + 2$.*

## 3.5 Primitive Reed-Solomon Codes

A special subclass of RS codes are called *primitive RS* codes. For these we take $a \in \mathbb{F}_q^\star$ a primitive element and define the evaluation points $\alpha \in \mathbb{F}_q^n$ of the RS code as

$$\alpha_i = a^i$$

for all $i \in \{0, \ldots, q-2\}$.

**Definition 3.22.** Let $q$ be a prime power and $k \leq n = q-1$ be positive integers and $a \in \mathbb{F}_q^\star$ a primitive element. Further, let $\alpha = (1, a, a^2, \ldots, a^{q-2}) \in \mathbb{F}_q^n$. The code $\mathcal{RS}_{q,n,k}(\alpha)$ is called *primitive RS* code.

Primitive RS codes have as a generator matrix

$$V_{q,n,k}(\alpha) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & a & \cdots & a^{q-2} \\ \vdots & \vdots & & \vdots \\ 1 & a^{k-1} & \cdots & a^{(k-1)(q-2)} \end{pmatrix}.$$

**Lemma 3.23.** *Let $q$ be a prime power, then*

$$\sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell = \begin{cases} 0 & \text{if } (q-1) \nmid \ell, \\ -1 & \text{if } (q-1) \mid \ell. \end{cases}$$

*Proof.* If $(q-1) \mid \ell$, then there exists a positive integer $m$ such that $m(q-1) = \ell$ and

$$\sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell = \sum_{\alpha \in \mathbb{F}_q^\star} \alpha^{m(q-1)} = \sum_{\alpha \in \mathbb{F}_q^\star} (\alpha^{q-1})^m = \sum_{\alpha \in \mathbb{F}_q^\star} 1 = q-1.$$

On the other hand, if $(q-1) \nmid \ell$, then for any primitive element $a \in \mathbb{F}_q^\star$, we have that $a^\ell \neq 1$. Multiplying by $a$ introduces a bijection $\varphi_a : \mathbb{F}_q^\star \to \mathbb{F}_q^\star, \alpha \mapsto a\alpha$. Thus,

$$\sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell = \sum_{\alpha \in \mathbb{F}_q^\star} (a\alpha)^\ell = a^\ell \sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell.$$

Since $a^\ell \neq 1$, we must have $\sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell = 0$. □

We may also extend the primitive RS code to consider $n = q$, i.e., the evaluation points are all the elements of the finite field, that is let $\alpha = (0, 1, a, a^2, \ldots, a^{q-2}) \in \mathbb{F}_q^n$ and consider the code $\mathcal{RS}_{q,n,k}(\alpha)$.

A generator matrix for $\mathcal{RS}_{q,q,k}(\alpha)$ is given by

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & a & \cdots & a^{q-2} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & a^{k-1} & \cdots & a^{(q-2)(k-1)} \end{pmatrix}.$$

In the case of this length $q$ RS code, we have that their duals are again RS codes.

**Proposition 3.24.** *Let $q$ be a prime power and $k \leq n = q$ be positive integers and $a \in \mathbb{F}_q^\star$ a primitive element. Further, let $\alpha = (0, 1, a, a^2, \ldots, a^{q-2}) \in \mathbb{F}_q^n$. Then $\mathcal{RS}_{q,n,k}^\perp(\alpha) = \mathcal{RS}_{q,n,n-k}(\alpha)$.*

**Exercise 3.25.** *Prove Proposition 3.24 using Lemma 3.23.*

**Proposition 3.26.** *Let $q$ be a prime power and $k \leq n = q$ be positive integers and $a \in \mathbb{F}_q^\star$ a primitive element. Further, let $\alpha = (0, 1, a, a^2, \ldots, a^{q-2}) \in \mathbb{F}_q^n$. Then $\mathcal{RS}_{q,n,k}(\alpha)$ is an MDS code.*

**Exercise 3.27.** *Prove Proposition 3.26 using again that any polynomial $f(x) \in \mathbb{F}_q[x]$ of degree $\ell < k$ can have at most $\ell$ roots in $\mathbb{F}_q$.*

We can even consider an "extended" RS code to get to the length $n = q + 1$, by considering

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 0 \\ 0 & 1 & a & \cdots & a^{q-2} & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & a^{k-1} & \cdots & a^{(q-2)(k-1)} & 1 \end{pmatrix}.$$

**Proposition 3.28.** *Let $q$ be a prime power and $k \leq n = q + 1$ be positive integers and $a \in \mathbb{F}_q^\star$ a primitive element. Then $\mathcal{RS}_{q,n,k}(\alpha) = \langle G \rangle$, where*

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 0 \\ 0 & 1 & a & \cdots & a^{q-2} & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & a^{k-1} & \cdots & a^{(q-2)(k-1)} & 1 \end{pmatrix}$$

*is an MDS code.*

*Proof.* Recall that any $c \in \mathcal{RS}_{q,n-1,k}(\alpha)$ is of the form

$$c = (f(0), f(1), f(a), \ldots, f(a^{q-2})),$$

where $f(x) = \sum_{i=0}^{k-1} f_i x^i$ for some $f_i \in \mathbb{F}_q$. We can send $c$ to $\varphi(c) \in \mathcal{RS}_{q,n,k}(\alpha)$ by defining

$$\varphi(c) = (f(0), f(1), f(a), \ldots, f(a^{q-2}), f_{k-1}).$$

After noting that, we can proceed as in the usual proof, that is $f(x) \in \mathbb{F}_q[x]$ of degree $\ell$ can have at most $\ell$ roots. If the last entry $f_{k-1} = 0$, then we are in the case of degree $\ell < k - 1$ and thus can have at most $< k - 1$ zeros in the first $q$ positions. □

**Example 3.29.** *Let us consider $q = 3, k = 2$ and $\alpha = (0, 1, 2)$. Then $\mathcal{RS}_{3,3,2}(\alpha) = \langle G \rangle$, where*

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

*The codeword associated to the polynomial* $f(x) = 1$ *is* $c = (1, 0)G = (1, 1, 1)$ *and for* $f'(x) = x$ *we get* $c' = (0, 1)G = (0, 1, 2)$. *By the extension to* $\mathcal{RS}_{3,4,2}(\alpha)$ *we get that* $\varphi(c) = (1, 1, 1, 0)$ *as* $f_1 = 0$ *and* $\varphi(c') = (0, 1, 2, 1)$ *as* $f_1' = 1$ *and thus*

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

*is such that* $\langle \varphi(G) \rangle = \mathcal{RS}_{3,4,2}(\alpha)$.

Can we have even larger length $n$? According to the famous MDS conjecture, we cannot.

## 3.6 MDS Conjecture

We have seen that Reed-Solomon codes are MDS codes, but unfortunately they have a length which is bounded from above by the size of the finite field. The existence of longer MDS codes is still an open problem: Apart from Reed-Solomon codes, the only "long" MDS codes which are known are the trivial MDS codes, $[n, n]_q$ and $[n, n-1]_q$, and the following special case in characteristic two:

**Proposition 3.30.** *Let* $q = 2^m$, *for some positive integer* $m$. *There exists an MDS code of length* $q + 2$ *and dimension 3 and an MDS code of length* $q + 2$ *and dimension* $q - 1$ *(which is its dual).*

We show this existence only in a small case, namely over $\mathbb{F}_4$.

**Example 3.31.** *Consider the finite field* $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$, *where* $\alpha^2 = \alpha + 1$. *The code generated by*

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

*is a* $[6, 3]_4$ *linear MDS code. In fact, we can easily check that any* $3 \times 3$ *submatrix of* $G$ *is invertible.*

The history of the MDS conjecture is quite long and old, in fact it is older than the Singleton bound itself! This is due to the connection of coding theory to finite geometry. We will not go deeper into this connection in this course, but in case you are familiar with finite geometry, MDS codes correspond to arcs in $PG(k-1, q)$, RS codes correspond to normal rational curves and finally the MDS conjecture states that an arc cannot be larger than a normal rational curve. This conjecture is due to Segre [16].

**Conjecture 3.32** (Segre, 1955)**.** *Let* $q$ *be a prime power and* $k \leq n$ *be positive integers. Then any* $[n, k]_q$ *linear MDS code is such that* $n \leq q + 1$, *except for*

- *the trivial MDS codes* $[n, n]_q$ *and* $[n, n-1]_q$,
- *the exceptional MDS codes in Proposition 3.30 in characteristic 2.*

The conjecture is widely believed to be true, and some cases are also known to be true, most importantly, over prime fields [2].

**Theorem 3.33** (Ball, 2012)**.** *The MDS conjecture is true if $q$ is a prime.*

Does that mean there are no other codes attaining the Singleton bound?

Absolutely not. There is a long list of constructions of MDS codes which are not GRS codes, however their length $n$ does not exceed $q+1$. Even more is true, if we fix the length and dimension and let $q$ grow, we expect random codes to attain the Singleton bound.

**Theorem 3.34.** *Let $k \leq n$ be positive integers. Then,*

$$\frac{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \textit{dim}(\mathcal{C}) = k, d_H(\mathcal{C}) = n-k+1\}|}{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \textit{dim}(\mathcal{C}) = k\}|} \xrightarrow[q\to\infty]{} 1.$$

*Proof.* Recall that the number of codes of dimension $k$ and length $n$ is given by

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \prod_{i=0}^{k-1} \frac{q^n - q^i}{q^k - q^i}.$$

To show that the ratio of MDS codes tends to 1, it is enough to show that a lower bound $\ell(q)$ tends to one, as

$$1 \geq \frac{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k, d_H(\mathcal{C}) = n-k+1\}|}{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k\}|} \geq \ell(q) \xrightarrow[q\to\infty]{} 1.$$

In order to lower bound the number of MDS codes, we use the characterization, that every $k \times k$ submatrix of a generator matrix $G$ is invertible. Thus, we start with a full rank $k \times k$ matrix, which are

$$\prod_{i=0}^{k-1} (q^k - q^i)$$

many. For the remaining $n-k$ columns, we choose vectors in $\mathbb{F}_q^k$ which do not lie in the span of any $k-1$ vertices among the ones already picked. At each step $i \in \{0, \ldots, n-k-1\}$, we have $\binom{k+i}{k-1}$ choices to build a subspace of dimension $k-1$, thus this leads to

$$\prod_{i=0}^{k-1} (q^k - q^i) \prod_{i=0}^{n-k-1} \left( q^k - q^{k-1} \binom{k+i}{k-1} \right).$$

This is not the exact number of MDS codes, as some subspaces can also intersect, hence by subtracting $q^{k-1}$ for all possible choices of columns, we subtract too much, getting a lower bound. Clearly, different matrices $G$ will lead to the same code, in fact any $S \in \mathrm{GL}_q(k)$ is such that $SG$ is again a generator matrix. Thus

$$|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k, d_H(\mathcal{C}) = n-k+1\}| \geq \frac{\prod_{i=0}^{k-1}(q^k - q^i) \prod_{i=0}^{n-k-1} \left(q^k - q^{k-1}\binom{k+i}{k-1}\right)}{\prod_{i=0}^{k-1}(q^k - q^i)}.$$

Thus, the probability of having a MDS code is bounded from below by

$$\frac{\prod_{i=0}^{n-k-1}\left(q^k - q^{k-1}\binom{k+i}{k-1}\right)}{\prod_{i=0}^{k-1}\frac{q^n-q^i}{q^k-q^i}} = \frac{q^{kn}\prod_{i=0}^{k-1}(1-q^{i-k})\prod_{i=0}^{n-k-1}\left(1-q^{-1}\binom{k+i}{k-1}\right)}{q^{kn}\prod_{i=0}^{k-1}(1-q^{i-n})}$$
$$= \frac{\prod_{i=0}^{k-1}(1-q^{i-k})\prod_{i=0}^{n-k-1}\left(1-q^{-1}\binom{k+i}{k-1}\right)}{\prod_{i=0}^{k-1}(1-q^{i-n})}.$$

Note that $\binom{k+i}{k-1} \leq \binom{k+n-k-1}{k-1} \leq 2^n$ is a constant for growing $q$, and

$$\frac{\prod_{i=0}^{k-1}(1-q^{i-k})}{\prod_{i=0}^{k-1}(1-q^{i-n})} = \begin{bmatrix} n \\ k \end{bmatrix}_{1/q}^{-1} \xrightarrow[q\to\infty]{} 1.$$

Finally,

$$\frac{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k, d_H(\mathcal{C}) = n-k+1\}|}{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k\}|}$$
$$\geq \frac{\prod_{i=0}^{n-k-1}(1-2^nq^{-1})}{\begin{bmatrix} n \\ k \end{bmatrix}_{1/q}}$$
$$\geq \frac{(1-2^nq^{-1})^{n-k}}{\begin{bmatrix} n \\ k \end{bmatrix}_{1/q}} \xrightarrow[q\to\infty]{} 1.$$

□

On the other hand, if we fix the alphabet size $q$ and the rate $R$ and let $n$ grow, then MDS codes have density 0. One could easily prove this assuming the MDS conjecture, however, it also holds without assuming the conjecture is true.

**Theorem 3.35.** *Let $q$ be a prime power and $R \in (0,1)$. For $n \in \mathbb{N}$, let $k_n = \lfloor Rn \rfloor$. Then,*

$$\frac{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \textit{dim}(\mathcal{C}) = k_n, d_H(\mathcal{C}) = n-k_n+1\}|}{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \textit{dim}(\mathcal{C}) = k_n\}|} \xrightarrow[n\to\infty]{} 0.$$

*Proof.* The proof works similarly as for Theorem 3.34. This time we need an upper bound $u(n)$ which tends to 0, as

$$0 \leq \frac{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k_n, d_H(\mathcal{C}) = n-k_n+1\}|}{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k_n\}|} \leq u(n) \xrightarrow[n\to\infty]{} 0.$$

To get an upper bound, we thus want to subtract too little, e.g. only once $q^{k_n-1}$, assuming any choice of $k_n - 1$ columns spans the same subspace, leading to

$$|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k_n, d_H(\mathcal{C}) = n-k_n+1\}|$$
$$\leq \frac{\prod_{i=0}^{k_n-1}(q^{k_n}-q^i)\prod_{i=0}^{n-k_n-1}\left(q^{k_n}-q^{k_n-1}\right)}{\prod_{i=0}^{k_n-1}(q^{k_n}-q^i)}.$$

Thus, by dividing again by $\begin{bmatrix} n \\ k_n \end{bmatrix}_q$ we get

$$\begin{aligned}
&\frac{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k_n, d_H(\mathcal{C}) = n - k_n + 1\}|}{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k_n\}|} \\
&\leq \frac{\prod_{i=0}^{k_n-1}(q^{k_n} - q^i) \prod_{i=0}^{n-k_n-1} \left(q^{k_n} - q^{k_n-1}\right)}{\prod_{i=0}^{k_n-1}(q^n - q^i)} \\
&= \frac{\prod_{i=0}^{n-k_n-1} (1 - q^{-1})}{\begin{bmatrix} n \\ k_n \end{bmatrix}_{1/q}},
\end{aligned}$$

similar to the previous proof.

We note that

$$\begin{bmatrix} n \\ k_n \end{bmatrix}_{1/q} \xrightarrow[n\to\infty]{} \prod_{i=1}^{\infty}(1 - q^{-i})^{-1} \geq 1/4.$$

Then

$$\begin{aligned}
&\frac{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k_n, d_H(\mathcal{C}) = n - k_n + 1\}|}{|\{\mathcal{C} \subset \mathbb{F}_q^n \mid \dim(\mathcal{C}) = k_n\}|} \\
&\leq \prod_{i=0}^{n-k_n-1} \left(1 - q^{-1}\right) \begin{bmatrix} n \\ k_n \end{bmatrix}_{1/q}^{-1} \\
&\leq 4(1 - q^{-1})^{n-k_n} \xrightarrow[n\to\infty]{} 0.
\end{aligned}$$

□

We can also state the Singleton bound in its asymptotic form. Given a function $f(n)$, computing the asymptotic form of $f(n)$ means to compute

$$\lim_{n\to\infty} \frac{1}{n} \log_q(f(n)),$$

if it exists.

In the case of the Singleton bound, i.e., $d \leq n - k + 1$, we thus see $k$ and $d$ as functions in $n$, and assume that

$$\lim_{n\to\infty} \frac{k(n)}{n} = R, \quad \lim_{n\to\infty} \frac{d(n)}{n} = \delta$$

exist.

Note that $d/n$ is called the *relative minimum distance* and $R$ usually denotes the rate. In the asymptotic case, we de not make a difference whether $n$ grows or is fixed.

**Corollary 3.36.** *Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code and assume that $k$ and $d$ are functions in $n$, while $q$ is fixed. Then,*

$$\delta \leq 1 - R + o(1).$$

A different way to phrase such asymptotic result, is stating that there must exist a sequence of codes, whose parameters tend to this limit.

**Corollary 3.37.** *Let $(\mathcal{C}_r)_{r\in\mathbb{N}}$ be a sequence of codes with parameters $[n_r, k_r, d_r]$ over $\mathbb{F}_q$. Assume that*

$$R = \lim_{r\to\infty} \frac{k_r}{n_r}, \quad \delta = \lim_{r\to\infty} \frac{d_r}{n_r}$$

*exist. Then*

$$R + \delta \leq 1.$$

## 3.7 Decoding of RS Codes

Throughout this section, we let $q$ be a prime power, $k \leq n$ be positive integers and $\alpha \in \mathbb{F}_q^n$ be such that $\alpha_i$ are pairwise distinct. Recall that any codeword $c \in \mathcal{RS}_{q,n,k}(\alpha)$ is given as

$$c = (f(\alpha_1), \ldots, f(\alpha_n))$$

for some $f(x) \in \mathbb{F}_q[x]$ of degree $< k$.

Recall that the main aim of codes is to correct errors and RS codes can correct the most errors for a given length $n$ and dimension $k$, namely

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor.$$

A decoder is given a received word $r = c + e$, where $c = mG$ is a codeword of $\mathcal{C} = \langle G \rangle$ and $e$ is an error vector of weight up to $t$. The decoder then returns either $e, c$ or $m$.

**Exercise 3.38.** *Show that given any of the three vectors $e, c, m$ and knowing $G$, one can easily recover the other vectors.*

Note that any decoder of a RS code $\mathcal{RS}_{q,n,k}(\alpha)$ will also decode a GRS code $\mathcal{GRS}_{q,n,k}(\alpha, \beta)$. In fact, assume we have received $r = c+e$ with $c \in \mathcal{GRS}_{q,n,k}(\alpha, \beta)$ and $e \in \mathbb{F}_q^n$ of weight $\mathrm{wt}_H(e) \leq t = \lfloor \frac{n-k}{2} \rfloor$. Note that $c' = (c_1\beta_1^{-1}, \ldots, c_n\beta_n^{-1}) \in \mathcal{RS}_{q,n,k}(\alpha)$. Hence we can simply compute $r' = (r_1\beta_1^{-1}, \ldots, r_n\beta_n^{-1})$ and a decoder for $\mathcal{RS}_{q,n,k}(\alpha)$ would find $c'$ and knowing $\beta_1, \ldots, \beta_n$, we can easily recover $c \in \mathcal{GRS}_{q,n,k}(\alpha, \beta)$.

One of the most famous ones, and easy to understand is the *Berlekamp-Welch decoder.*

### 3.7.1 Berlekamp-Welch

Let us denote by $\mathrm{supp}_H(e) = \{i \in \{1, \ldots, n\} \mid e_i \neq 0\}$ the support of $e$. Let $c \in \mathcal{RS}_{q,n,k}(\alpha)$ be such that $c = (f(\alpha_1), \ldots, f(\alpha_n))$ for some $f(x) \in \mathbb{F}_q[x]$ of degree $< k$, and let $r = c + e$.

Let us define the error polynomial as

$$E(x) = \prod_{i \in \mathrm{supp}_H(e)} (x - \alpha_i).$$

Since $|\text{supp}_H(e)| \leq t$, the polynomial $E$ has degree at most $t$.

Note that for any $i \in \text{supp}_H(e)$, we have that $E(\alpha_i) = 0$, while for $i \notin \text{supp}_H(e)$, we get $E(\alpha_i) \neq 0$.

With this we immediately get that

$$r_i E(\alpha_i) = f(\alpha_i)E(\alpha_i)$$

for all $i \in \{1, \ldots, n\}$. This gives a system of $n$ equations, where the unknowns are the coefficients of $E(x)$ and $f(x)$, and the system is not linear. To solve this issue, we want to linearize it and hence define

$$N(x) = E(x)f(x).$$

As $\deg(f) \leq k-1, \deg(E) \leq t$, we get $\deg(N) \leq k+t-1$.

The system now becomes

$$r_i E(\alpha_i) = N(\alpha_i), \tag{1}$$

for all $i \in \{1, \ldots, n\}$. Thus, it is a linear system of $n$ equations and the unknowns, being the coefficients of $E(x)$ and $N(x)$, are $k + 2t + 1 \leq n + 1$ many. In fact, a degree $\ell$ polynomial has $\ell + 1$ coefficients. We have the existence of a non-trivial solution (namely the coefficients of $E(x)$ and $N(x)$). Moreover any other non-trivial solution allows us to recover $f(x)$.

**Lemma 3.39.** *Let $(E_1(x), N_1(x)), (E_2(x), N_2(x)) \in \mathbb{F}_q[x]^2$ of degree $\deg(E_i) \leq t, \deg(N_i) \leq k+t-1$ for $i \in \{1,2\}$ be two distinct non-trivial solutions of* (1)*. Then $E_1(x), E_2(x) \neq 0$ and*

$$\frac{N_1(x)}{E_1(x)} = \frac{N_2(x)}{E_2(x)} = f(x).$$

*Proof.* If we set $E_1(x) = 0$ in the system (1), we get the system

$$0 = N(\alpha_i)$$

for all $i \in \{1, \ldots, n\}$. Thus, $N(x)$ has $n$ distinct roots $\alpha_i$ and since its degree is at most $k+t-1 < n$, we must have $N(x) = 0$ and thus the solution is trivial. Clearly, the same holds for $E_2(x)$ and we get the first claim, i.e., $E_1(x), E_2(x) \neq 0$.

Set $S(x) = N_1(x)E_2(x) - N_2(x)E_1(x)$, with

$$\deg(S) \leq k + 2t - 1 \leq n - 1.$$

Since $(E_1(x), N_1(x)), (E_2(x), N_2(x))$ are both solutions to (1), we also have that

$$\begin{aligned} S(\alpha_i) &= N_1(\alpha_i)E_2(\alpha_i) - N_2(\alpha_i)E_1(\alpha_i) \\ &= r_i E_1(\alpha_i)E_2(\alpha_i) - r_i E_2(\alpha_i)E_1(\alpha_i) = 0, \end{aligned}$$

for all $i \in \{1, \ldots, n\}$. Thus $S$ has $n$ distinct roots $\alpha_i$ but since its degree is less than $n$, it must be the zero polynomial and thus $N_1(x)E_2(x) = N_2(x)E_1(x)$ for every non-trivial solution, in particular also for $N(x) = E(x)f(x)$, which leads to the second claim

$$\frac{N_1(x)}{E_1(x)} = f(x).$$

□

Thus, we only have to solve the system (1) and given a non-trivial solution $(E(x), N(x))$ compute $f(x) = N(x)/E(x)$. The complexity of this algorithm is thus in $\mathcal{O}(n^3)$.

We can also invoke a check, namely $\deg(f) < k$, and

$$d_H(r, (f(\alpha_1), \ldots, f(\alpha_n))) \leq t.$$

If this is not fulfilled, the weight of the error vector $e$ is larger than the error-correction capability. We will later see how to decode RS codes beyond this threshold.

**Example 3.40.** *Let us consider $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ where $\alpha^2 = \alpha + 1$. Then $\mathcal{RS}_{4,4,2}(0, 1, \alpha, 1 + \alpha)$ can correct 1 error. Assume we received $r = (1, 1 + \alpha, \alpha, \alpha)$.*

*We know that $\deg(E) \leq 1$ and $\deg(N) \leq 2$, thus $E(x) = e_0 + e_1 x, N(x) = n_0 + n_1 x + n_2 x^2$ and our unknowns are $e_0, e_1, n_0, n_1, n_2$.*

*From the equations $r_i E(\alpha_i) - N(\alpha_i) = 0$ we build the following linear system*

$$\begin{pmatrix} r_1 & r_1\alpha_1 & -1 & -\alpha_1 & -\alpha_1^2 \\ r_2 & r_2\alpha_2 & -1 & -\alpha_2 & -\alpha_2^2 \\ r_3 & r_3\alpha_3 & -1 & -\alpha_3 & -\alpha_3^2 \\ r_4 & r_4\alpha_4 & -1 & -\alpha_4 & -\alpha_4^2 \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ n_0 \\ n_1 \\ n_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1+\alpha & 1+\alpha & 1 & 1 & 1 \\ \alpha & 1+\alpha & 1 & \alpha & 1+\alpha \\ \alpha & 1 & 1 & 1+\alpha & \alpha \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ n_0 \\ n_1 \\ n_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

*By solving this system (e.g. using Gaussian elimination) we get the solution*

$$(e_0, e_1, n_0, n_1, n_2) = (1 + \alpha, 1, 1 + \alpha, 0, \alpha)$$

*and we see that*

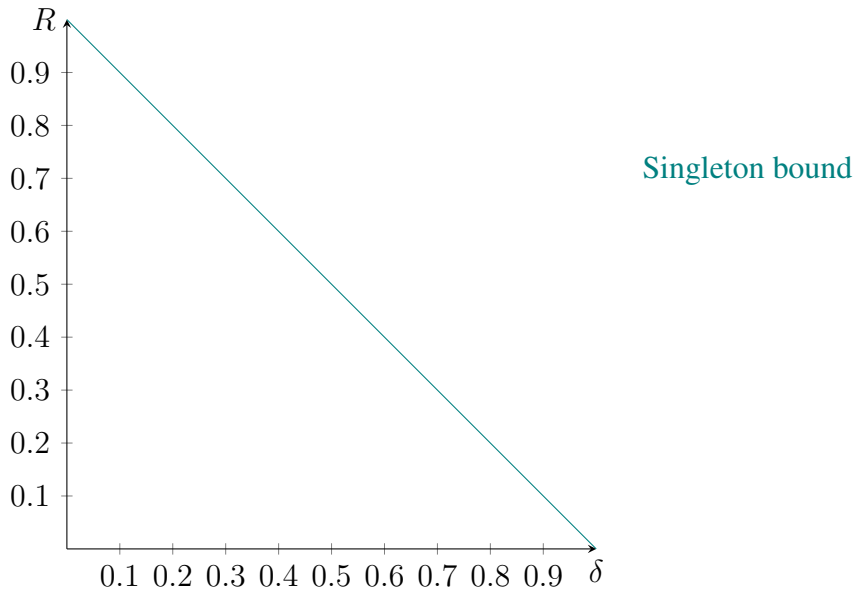
$$\frac{N(x)}{E(x)} = \frac{1 + \alpha + \alpha x^2}{1 + \alpha + x} = 1 + \alpha x = f(x).$$

*Hence the sent codeword is*

$$c = (f(\alpha_1), f(\alpha_2), f(\alpha_3), f(\alpha_4)) = (1, 1 + \alpha, \alpha, 0)$$

*which has indeed distance 1 from $r$.*

- The *Singleton bound* states that every $[n, k, d]$ code is such that $d \leq n - k + 1$.
- Codes attaining this bound are called *MDS*.
- $\mathcal{C}$ is MDS if and only if $\mathcal{C}^{\perp}$ is MDS.
- Asymptotically, the bound states $\delta \leq 1 - R$.
- *Generalized Reed-Solomon* codes are MDS.
- The GRS construction requires $n \leq q + 1$.
- It is unknown whether larger non-trivial MDS codes exist.
- For fixed $k \leq n$ and $q$ growing, the probability of an $[n, k]_q$ code being MDS goes to one.
- For fixed $q$ and $n$ growing, the probability of an $[n, \lfloor Rn \rfloor]_q$ code being MDS goes to zero.



# 4 Sphere-Packing and Sphere-Covering

Recall that the Singleton bound gives an upper bound on the size of a code with prescribed minimum distance $d_H(\mathcal{C})$ and length $n$ as

$$|\mathcal{C}| \leq q^{n-d_H(\mathcal{C})+1}.$$

Apart from this upper bound, we also have the sphere-packing bound, also called *Hamming bound* and a lower bound given by the sphere-covering bound, also called *Gilbert-Varshamov bound*.

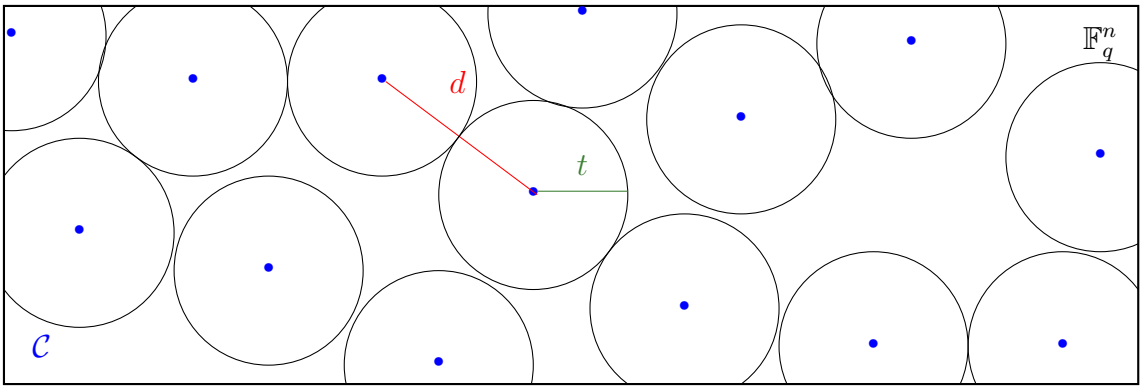
Let us denote by $V_H(r,n,q)$ the size (or volume) of a ball, i.e.,

$$V_H(r,n,q) = |B_H(r,n,q,x)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i$$

for any $x \in \mathbb{F}_q^n$.

## 4.1 Hamming bound

The Hamming bound, or sphere-packing bound can be depicted as follows:




In this picture we cannot place any other codeword, without their balls of radius $t$ intersecting.

**Theorem 4.1** (Hamming Bound)**.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k]_q$ code with $d_H(\mathcal{C}) \geq d$ and let $t = \left\lfloor \frac{d-1}{2} \right\rfloor$. Then*

$$|\mathcal{C}| \leq \frac{q^n}{V_H(t,n,q)}.$$

*Proof.* The balls of radius $t$ around codewords of $\mathcal{C}$ have to be pairwise disjoint. In fact, if there exist $c \neq c' \in \mathcal{C}$ such that

$$x \in B_H(t,n,q,c) \cap B_H(t,n,q,c')$$

for some $x \in \mathbb{F}_q^n$, then by the triangle inequality

$$d_H(c, c') \leq d_H(x, c) + d_H(x, c') \leq 2t < d \leq d_H(\mathcal{C}),$$

a contradiction to $d_H(\mathcal{C})$ being the minimum distance among all distinct codewords.

Thus, we must have that the union of all balls around the codewords is disjoint and contained in $\mathbb{F}_q^n$, i.e.,

$$\bigcup_{c \in \mathcal{C}} B_H(t, n, q, c) \subseteq \mathbb{F}_q^n$$

and due to the disjointness

$$|\mathcal{C}| V_H(t, n, q) = \sum_{c \in \mathcal{C}} V_H(t, n, q) = |\bigcup_{c \in \mathcal{C}} B_H(t, n, q, c)| \leq q^n.$$

□

We can also provide an alternative proof, which follows a greedy algorithm.

For this let us denote by $A(d, n, q)$ the largest size of *any* code (also non-linear), in $\mathbb{F}_q^n$ with minimum distance at least $d$. The Hamming bound then states that

$$A(d, n, q) \leq \frac{q^n}{V_H(t, n, q)},$$

and since any linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with minimum distance at least $d$ is such that $|\mathcal{C}| \leq A(d, n, q)$ we recover the Hamming bound from Theorem 4.1.

However, the proof now follows a greedy argument: we start by placing a random vector $c \in \mathbb{F}_q^n$ in our code $\mathcal{C}$. We then build a ball of radius $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ around $c$ and choose a next codeword $c'$ with the property

$$B_H(t, n, q, c) \cap B_H(t, n, q, c') = \emptyset.$$

This way, we ensure that $d_H(c, c') > 2t$.

We continue placing codewords and their balls of radius $t$, until there is no more space, getting a "packing", i.e., a disjoint union

$$\bigcup_{c \in \mathcal{C}} B_H(t, n, q, c) \subseteq \mathbb{F}_q^n.$$

The rest of the proof works in the same way.

## 4.2 Perfect Codes

We are again interested in codes which attain this bound.

**Definition 4.2.** Let $q$ be a prime power and $k \leq n$ be positive integers. An $[n, k, d]_q$ code $\mathcal{C}$ with

$$q^{n-k} = V_H\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n, q\right)$$

is called a *perfect* code.

They are called perfect, for a very special property: they can correct any received vector.

**Proposition 4.3.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n, k, d]_q$ code and set $t = \lfloor \frac{d-1}{2} \rfloor$. Then, $\mathcal{C}$ is perfect if and only if for all $r \in \mathbb{F}_q^n$ there exists a unique codeword $c \in \mathcal{C}$ such that*

$$r \in B_H(t, n, q, c).$$

*Proof.* For the first direction, we assume that $\mathcal{C}$ is a perfect code, i.e.,

$$|\mathcal{C}| = \frac{q^n}{V_H(t, n, q)}$$

which implies that

$$\bigcup_{c \in \mathcal{C}} B_H(t, n, q, c) = \mathbb{F}_q^n,$$

as a disjoint union. Thus, for any $r \in \mathbb{F}_q^n$ there exists a unique ball $B_H(t, n, q, c)$ such that $r \in B_H(t, n, q, c)$.

For the other direction, we have that any $r \in \mathbb{F}_q^n$ is in exactly one ball $B_H(t, n, q, c)$ and thus

$$\bigcup_{c \in \mathcal{C}} B_H(t, n, q, c) = \mathbb{F}_q^n$$

as disjoint union, implying that

$$|\mathcal{C}| = \frac{q^n}{V_H(t, n, q)}$$

and that $\mathcal{C}$ is perfect. □

**Example 4.4.** *Let $n$ be an odd positive integer and consider the $[n, 1]_2$ repetition code $\mathcal{C}$. Recall that $d_H(\mathcal{C}) = n = 2t + 1$, for some $t$ and thus,*

$$|\mathcal{C}| = 2 = \frac{2^n}{2^{n-1}} = \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}.$$

The fact, that we had to choose an odd minimum distance in this example is actually always true for perfect codes:

**Proposition 4.5.** *Let $q$ be a prime power and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n, k, d]_q$ code. If $d$ is even, then $\mathcal{C}$ is not perfect.*

*Proof.* Assume that $d = 2(t+1)$ for some $t$ and note that $t = \lfloor \frac{d-1}{2} \rfloor$. Let $c \in \mathcal{C}$ and $r \in \mathbb{F}_q^n$ be such that $d_H(c, r) = t + 1$.

To show that $\mathcal{C}$ is not perfect, it is enough to show that there is no codeword $c' \in \mathcal{C}$, such that $r \in B_H(t, n, q, c')$.

Assume by contradiction, that such $c' \in \mathcal{C}$ exists, then by the triangle inequality we get that

$$d_H(c, c') \leq d_H(c', r) + d_H(c, r) \leq t + t + 1 < d,$$

a contradiction to $d$ being the minimum distance. □

There only exist very few non-trivial perfect codes: the Golay code $[23, 12, 7]_2$ the Golay code $[11, 6, 5]_3$ and the *Hamming code.*

**Definition 4.6.** Let $q$ be a prime power and $r \geq 2$ be a positive integer. Let $n = \frac{q^r-1}{q-1}$ and $H$ be the $r \times n$ matrix having as columns all vectors in $\mathbb{F}_q^r$ up to non-zero scalar multiples. Then $\mathcal{C} = \ker(H^\top)$ is called *Hamming code.*

**Example 4.7.** *Let us consider $q = 2$ and $r = 3$ and define*

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

*The code $\mathcal{C} = ker(H^\top)$ is a $[7, 4]_2$ Hamming code.*

Hamming codes are in fact perfect codes.

**Proposition 4.8.** *Let $q$ be a prime power and $r \geq 2$ be a positive integer and set $n = \frac{q^r-1}{q-1}$. The $[n, n-r]_q$ Hamming code has $d_H(\mathcal{C}) = 3$ and is perfect.*

*Proof.* Let us first show that the construction of a Hamming code provides us with the parameters $n = \frac{q^r-1}{q-1}$ and $k = n - r$. We note that there exist $q^r - 1$ non-zero vectors in $\mathbb{F}_q^r$ and by excluding scalar multiples (which are $q - 1$ many) we get that there are $n = \frac{q^r-1}{q-1}$ many vectors in $\mathbb{F}_q^r$ up to scalar multiples, and thus $\mathcal{C}$ has length $n$. As the vectors $e_i \in \mathbb{F}_q^r$ with only the $i$th entry being 1 and the rest 0, are choices (up to scalar multiplication) for columns of $H$, we get that $\text{rk}(H) = r$ and hence $\dim(\mathcal{C}) = n - r$.

We now prove that $d_H(\mathcal{C}) = 3$. Since one column is not a scalar multiple of another column, any two columns must be linearly independent. Moreover, there exist 3 columns which are linearly dependent (for example choosing $e_1, e_2$ and $e_1 + e_2$ again up to scalar multiples). Thus, $d_H(\mathcal{C}) = 3$ by Proposition 2.38.

We can now show that Hamming codes are perfect. For this we note that $t = 1$ and

$$V_H(t, n, q) = \sum_{i=0}^{1} \binom{n}{i} (q-1)^i = 1 + n(q-1) = 1 + q^r - 1 = q^r.$$

Hence,

$$|\mathcal{C}| = q^{n-r} = \frac{q^n}{q^r}.$$

□

The dual code of the Hamming code is called *simplex* code.

**Definition 4.9** (Simplex Code)**.** Let $q$ be a prime power and $r \geq 2$ be a positive integer. Let $n = \frac{q^r-1}{q-1}$ and $G$ be the $r \times n$ matrix having as columns all vectors in $\mathbb{F}_q^r$ up to non-zero scalar multiples. Then $\mathcal{C} = \langle G \rangle$ is called *simplex code.*

**Corollary 4.10.** *Let $q$ be a prime power and $r \geq 2$ be a positive integer. Let $n = \frac{q^r-1}{q-1}$. The simplex code is a $[n, r]_q$ code.*

We will see the simplex code again, after we introduced the Plotkin bound.

Note, often in literature the Hamming code and simplex code are only defined for $r \geq 3$. This is not because the Hamming code for $r = 2$ is not a perfect code, instead it is the known $[q + 1, q - 1]_q$ code. For example, for $q = 2$, we get the $[3, 1]_2$ repetition code. For $q = 3$, the simplex and Hamming codes coincide being a $[4, 2]_3$ code.

## 4.3 Asymptotic Hamming bound

Recall that given a function $f(n)$, computing the asymptotic form of $f(n)$ means to compute

$$\lim_{n\to\infty} \frac{1}{n} \log_q(f(n)),$$

if it exists.

We again let the dimension $k$ and the minimum distance $d$ be functions in $n$ and assume

$$R = \lim_{n\to\infty} \frac{k(n)}{n}, \quad \delta = \lim_{n\to\infty} \frac{d(n)}{n}$$

exist.

To give the asymptotic version of the Hamming bound, we first have to find

$$\lim_{n\to\infty} \frac{1}{n} \log_q(V_H(rn, n, q)),$$

for some $r \in [0, 1 - 1/q]$.

**Definition 4.11** (Entropy Function)**.** For a positive integer $q \geq 2$ the $q$-ary entropy function is defined as follows:

$$\begin{aligned} H_q : [0, 1] &\to \mathbb{R}, \\ x &\to x \log_q(q - 1) - x \log_q(x) - (1 - x) \log_q(1 - x). \end{aligned}$$

The entropy function is originally introduced in information theory, in particular setting $q = 2$, the entropy function $H_2(x)$ measures the uncertainty in the outcome of a $x$-biased coin toss (which lands heads with probability $x$ and tails with probability $1 - x$).

**Lemma 4.12.** *Let $q \geq 2$ and $n$ a positive integer and $x \in [0, 1 - 1/q]$. Then,*

$$V_H(xn, n, q) \leq q^{H_q(x)n}.$$

*Proof.* We have that

$$\begin{aligned}
\frac{V_H(xn, n, q)}{q^{H_q(x)n}} &= \frac{\sum_{i=0}^{xn} \binom{n}{i}(q-1)^i}{(q-1)^{xn}x^{-xn}(1-x)^{-(1-x)n}} \\
&= \sum_{i=0}^{xn} \binom{n}{i}(q-1)^i(q-1)^{-xn}x^{xn}(1-x)^{(1-x)n} \\
&= \sum_{i=0}^{xn} \binom{n}{i}(q-1)^i(1-x)^n \left(\frac{x}{(q-1)(1-x)}\right)^{xn}.
\end{aligned}$$

Since $x \leq 1 - 1/q$, we get that $x/(q-1) \leq 1/q \leq 1 - x$ and hence $\frac{x}{(q-1)(1-x)} < 1$ and by decreasing the power, we increase the formula. Thus,

$$\begin{aligned}
\frac{V_H(xn, n, q)}{q^{H_q(x)n}} &\leq \sum_{i=0}^{xn} \binom{n}{i}(q-1)^i(1-x)^n \left(\frac{x}{(q-1)(1-x)}\right)^{i} \\
&= \sum_{i=0}^{xn} \binom{n}{i}(1-x)^{n-i}x^i \\
&\leq \sum_{i=0}^{n} \binom{n}{i}(1-x)^{n-i}x^i = 1,
\end{aligned}$$

where we used the binomial theorem in the last equality. □

We can also give a lower bound, using Sterling's formula. For this, recall that

$$m! = \sqrt{2\pi m}\left(\frac{m}{e}\right)^m (1 + o(1))$$

and hence

$$\binom{n}{xn} \geq \left(\frac{1}{x}\right)^{xn} \left(\frac{1}{1-x}\right)^{(1-x)n} \exp(-o(n)) = 2^{H_q(x)n - o(n)}.$$

From this we can follow that

$$V_H(xn, n, q) \geq \binom{n}{xn}(q-1)^{xn} \geq q^{H_q(x)n - o(n)}.$$

**Corollary 4.13.** *Let $q \geq 2$ and $n$ a positive integer and $x \in [0, 1 - 1/q]$. Then,*

$$\lim_{n\to\infty} \frac{1}{n} \log_q(V_H(xn, n, q)) = H_q(x).$$

The asymptotic Hamming bound then states

**Corollary 4.14.** *Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code and assume that $k$ and $d$ are functions in $n$, while $q$ is fixed. Then,*

$$R \leq 1 - H_q(\delta/2) + o(1).$$

## 4.4 Gilbert-Varshamov Bound

We finally move to a first lower bound on the size of a code with prescribed length and minimum distance.

While the Hamming bound is considered a packing bound, the Gilbert-Varshamov bound is a covering bound.

Recall that $V_H(r, n, q)$ is the size (or volume) of a ball and $A(d, n, q)$ the largest size of *any* code (also non-linear), in $\mathbb{F}_q^n$ with minimum distance at least $d$.

We start with the non-linear version of the Gilbert-Varshamov bound, which provides a lower bound on $A(d, n, q)$.

**Theorem 4.15** (Gilbert-Varshamov Bound)**.** *Let $q$ be a prime power and $n, d$ be positive integers. Then,*

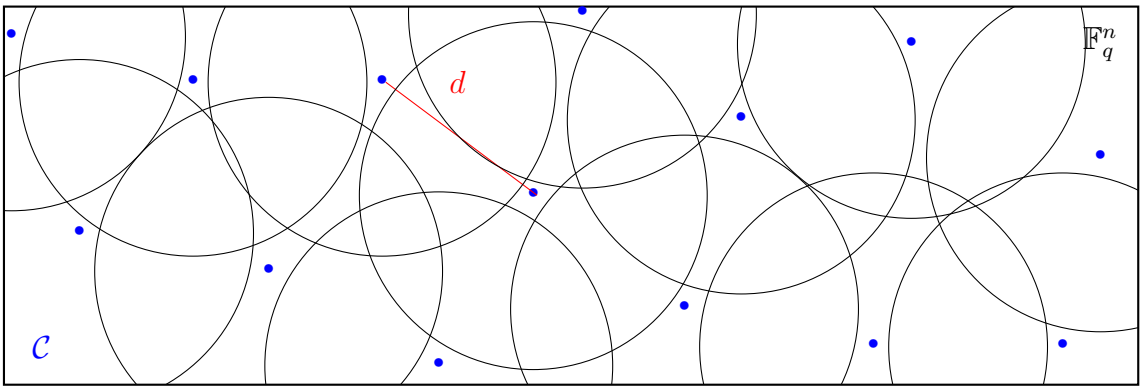
$$A(d, n, q) \geq \frac{q^n}{V_H(d-1, n, q)}.$$

Note that this bound is an existence bound. It should be read as follows:

*There exists a (possibly non-linear) code which has size larger than $\frac{q^n}{V_H(d-1,n,q)}$.*

This does however not imply that a given code should have size larger, equal or smaller than $\frac{q^n}{V_H(d-1,n,q)}$. In fact, many codes, e.g. RS codes have size smaller than $\frac{q^n}{V_H(d-1,n,q)}$, we will see later that random codes attain this bound with high probability, for large $n$ and finally, there exist also algebraic geometry codes which have a larger size.

We can again depict the idea of the Gilbert-Varshamov bound as



In this picture we cannot place any other codeword outside of any ball of radius $d-1$.

The proof of the non-linear version follows the same greedy argument as the proof for the Hamming bound.

*Proof.* We start by placing a random vector $c \in \mathbb{F}_q^n$ in our code $\mathcal{C}$. We then build a ball of radius $d-1$ around $c$ and choose a next codeword $c'$ outside of this ball, i.e., $c' \in \mathbb{F}_q^n \setminus B_H(d-1,n,q,c)$. This way, we ensure that $d_H(c,c') > d-1$.

We continue placing codewords and their balls of radius $d-1$, until there is no more space, getting a "covering" of $\mathbb{F}_q^n$, i.e.,

$$\mathbb{F}_q^n \subseteq \bigcup_{c\in\mathcal{C}} B_H(d-1,n,qc).$$

Hence, we get

$$q^n \leq |\bigcup_{c\in\mathcal{C}} B_H(d-1,n,qc)| \leq \sum_{c\in\mathcal{C}} V_H(d-1,n,q) = |\mathcal{C}|V_H(d-1,n,q).$$

□

There also exists a linear version of the Gilbert-Varshamov bound, which ensures the existence of a code with minimum distance at least $d$.

**Theorem 4.16** (Gilbert-Varshamov bound)**.** *Let $q$ be a prime power and let $k \leq n$ and $d$ be positive integers, such that*

$$V_H(d-2,n-1,q) < q^{n-k}.$$

*Then, there exists a $[n,k]_q$ linear code with minimum Hamming distance at least $d$.*

**Exercise 4.17.** *Prove Theorem 4.16 using a greedy algorithm to construct a $(n-k)\times n$ parity-check matrix $H$, such that every $d-1$ columns are linearly independent.*

## 4.5 Asymptotic Gilbert-Varshamov Bound

It turns out that random codes attain the asymptotic Gilbert-Varshamov (GV) bound with high probability for $n$ growing.

We again see $k, d$ as a function in $n$ and set

$$\delta = \lim_{n\to\infty} \frac{d(n)}{n} \in [0, 1-1/q], \quad R(\delta) = \limsup_{n\to\infty} \frac{1}{n} \log_q A(\delta n, n, q)$$

to be the relative minimum distance and the asymptotic information rate.

We can formulate the asymptotic Gilbert-Varshamov bound as the existence of a infinite family of codes as

**Theorem 4.18** (The Asymptotic Gilbert-Varshamov Bound)**.** *For every prime power $q$ and $\delta \in [0, 1 - 1/q]$ there exists an infinite family $\mathcal{C}$ of codes with rate*

$$R(\delta) \geq 1 - H_q(\delta).$$

**Theorem 4.19.** *Let $q$ be a prime power, $\delta \in [0, 1 - 1/q)$ and $0 < \varepsilon$ and $n$ a positive integer. Set $k \leq n(1 - H_q(\delta) - \varepsilon)$ and let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a random code of dimension $k$. Then, with high probability, $\mathcal{C}$ has minimum Hamming distance at least $\delta n$, for $n$ growing.*

*Proof.* We want to show that

$$\mathbb{P}(d_H(\mathcal{C}) > \delta n) \geq 1 - q^{-\varepsilon n}.$$

We first choose $G \in \mathbb{F}_q^{k \times n}$ uniform at random of rank $k$ and then bound the counter probability, that is

$$\mathbb{P}(d_H(\mathcal{C}) \leq \delta n)$$

which is given by the probability that there exists a non-zero codeword of weight at most $\delta n$. Since $G$ is uniform at random, also any non-zero codeword $mG \in \mathbb{F}_q^n \setminus \{0\}$ is uniform at random.

We note that for a random non-zero codeword the probability of having weight at most $\delta n$ can be bounded as

$$\mathbb{P}(\mathrm{wt}_H(mG) \leq \delta n) = \frac{V_H(\delta n, n, q)}{q^n - 1} \leq q^{n(H_q(\delta)-1)}.$$

Thus, using a union bound, we get

$$\begin{aligned}
\mathbb{P}(d_H(\mathcal{C}) \leq \delta n) &= \mathbb{P}(\exists m \in \mathbb{F}_q^k \setminus \{0\} : \mathrm{wt}_H(mG) \leq \delta n) \\
&\leq \sum_{m \in \mathbb{F}_q^k \setminus \{0\}} \mathbb{P}(\mathrm{wt}_H(mG) \leq \delta n) \\
&\leq (q^k - 1) q^{n(H_q(\delta)-1)} \\
&\leq q^{n(1-H_q(\delta)-\varepsilon)+n(H_q(\delta)-1)} = q^{-\varepsilon n}.
\end{aligned}$$

Hence $\mathbb{P}(d_H(\mathcal{C}) > \delta n) \geq 1 - q^{-\varepsilon n}$ and $\lim_{n \to \infty} \mathbb{P}(d_H(\mathcal{C}) > \delta n) = 1$. □

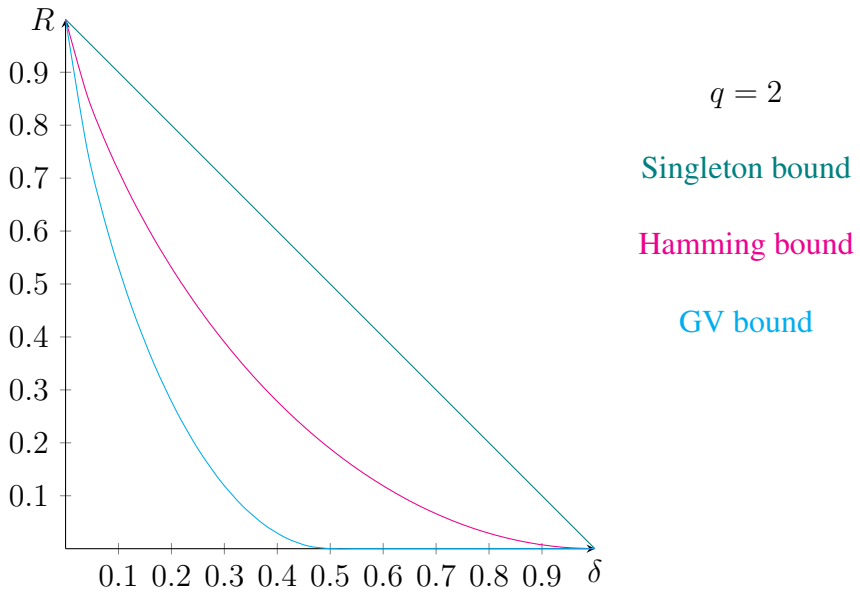
Hence, for large $q$, we know that random codes are with high probability MDS codes, i.e.,

$$d_H(\mathcal{C}) = n - k + 1,$$

while for large $n$, we now know that random codes attain with high probability the Gilbert-Varshamov bound, that is we may set

$$d_H(\mathcal{C}) = \max \left\{ r \;\middle|\; \sum_{i=0}^{r-1} \binom{n}{i} (q-1)^i < q^{n-k} \right\}.$$

- The *Hamming bound* states that every $[n, k, 2t+1]$ code is such that $V_H(t, n, q) \leq q^{n-k}$.
- Codes attaining this bound are called *perfect*.
- Asymptotically, the Hamming bound states $R \leq 1 - H_q(\delta/2)$.
- *Hamming* codes are perfect.
- The *Gilbert-Varshamov bound* states that there exists a (possibly non-linear) code with $|\mathcal{C}| \geq \frac{q^n}{V_H(d-1,n,q)}$.
- The linear GV bound states there there exists a linear $[n, k, d]$ code with $V_H(d-2, n-1, q) < q^{n-k}$.
- Asymptotically, the Gilbert-Varshamov bound states $R \geq 1 - H_q(\delta)$.
- For fixed $q$ and $n$ growing, the probability of an $[n, Rn]_q$ code attaining the GV bound goes to one.



# 5 Plotkin Bound

We come to one of our last bounds for the minimum distance $d_H(\mathcal{C})$ given $n, k$. This one is called Plotkin bound and again comes in different forms and with different proofs (we will see some of the proof techniques). It mainly comes from two easy observations:

1. All non-zero codewords have weight $\geq d_H(\mathcal{C})$. Thus, if we sum all their weights we must have
$$\sum_{c\in\mathcal{C}} \mathrm{wt}_H(c) = \sum_{c\in\mathcal{C}\setminus\{0\}} \mathrm{wt}_H(c) \geq (|\mathcal{C}|-1)d_H(\mathcal{C}),$$
with equality only if all non-zero codewords have minimal Hamming weight $d_H(\mathcal{C})$.
2. The average weight of a code is the same as the average weight of the whole space $\mathbb{F}_q^n$.

Let us elaborate more on the second point:

**Definition 5.1.** Let $\mathcal{C}$ be an $[n,k]_q$ linear code. The *average weight* of $\mathcal{C}$ is given by
$$\overline{\mathrm{wt}_H}(\mathcal{C}) = \frac{\sum_{c\in\mathcal{C}} \mathrm{wt}_H(c)}{|\mathcal{C}|}.$$

We can do a quick example to get an intuition.

**Example 5.2.** *Let us consider the code generated by*
$$G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix} \in \mathbb{F}_3^{2\times 3}.$$
*The code is then*
$$\mathcal{C} = \{(0,0,0),(1,0,1),(2,0,2),(0,1,2),(1,1,0),(2,1,1),(0,2,1),(1,2,2),(2,2,0)\}.$$
*We can quickly check that*
$$\overline{wt_H}(\mathcal{C}) = \frac{0+2+2+2+2+3+2+3+2}{3^2} = \frac{18}{9} = 2.$$

Let $i \in \{1,\dots,n\}$ and let us denote by $\pi_i$ the projection to the $i$th coordinate. That is
$$\begin{aligned} \pi_i : \mathbb{F}_q^n &\to \mathbb{F}_q \\ x = (x_1,\dots,x_n) &\mapsto x_i \end{aligned}$$

Then we can show that the amount of $x \in \mathbb{F}_q^n$ with $\pi_i(x) = a$ for a fixed $a \in \mathbb{F}_q$ is given by $q^{n-1}$.

**Lemma 5.3.** *Let $q$ be a prime power and $n$ a positive integer. Let $a \in \mathbb{F}_q$ and $i \in \{1, \ldots, n\}$. There are $q^{n-1}$ many $x \in \mathbb{F}_q^n$ such that $x_i = a$.*

*Or equivalently*

$$|\pi_i^{-1}(a)| = q^{n-1}.$$

*Proof.* We note that $\pi_i$ is a $\mathbb{F}_q$-linear map, since for any $x, x' \in \mathbb{F}_q^n, a \in \mathbb{F}_q$

$$\begin{aligned}\pi_i(x + x') &= \pi_i(x) + \pi_i(x'),\\ \pi_i(ax) &= a\pi_i(x).\end{aligned}$$

We also note that $\mathrm{Im}(\pi_i) = \mathbb{F}_q$, as it is an ideal in $\mathbb{F}_q$ and clearly not $\{0\}$. Finally, using the first isomorphism theorem, we get that

$$\mathbb{F}_q^n / \mathrm{ker}(\pi_i) \cong \mathbb{F}_q,$$

which also tells us that

$$|\mathrm{ker}(\pi_i)| = q^{n-1}.$$

With this we can compute $|\pi_i^{-1}(a)|$ as every $a \in \mathbb{F}_q$ has the same preimage size. □

**Exercise 5.4.** *Use a similar proof to show that the restriction of the projection on a non-degenerate $[n, k]_q$ linear code $\mathcal{C}$, i.e., $\pi_i : \mathcal{C} \to \mathbb{F}_q$ is also such that $|\pi_i^{-1}(a)| = q^{k-1}$.*

If we consider the whole ambient space $\mathbb{F}_q^n$, we can quickly compute its average weight.

**Lemma 5.5.** *Let $q$ be a prime power and $n$ be a positive integer. Then*

$$\overline{wt_H}(\mathbb{F}_q^n) = n\frac{q-1}{q}.$$

*Proof.* We observe that

$$\begin{aligned}\overline{\mathrm{wt}_H}(\mathbb{F}_q^n) &= \frac{\sum_{x \in \mathbb{F}_q^n} \mathrm{wt}_H(x)}{q^n}\\ &= \frac{\sum_{x \in \mathbb{F}_q^n} \sum_{i=1}^n \mathrm{wt}_H(x_i)}{q^n}\\ &= \frac{\sum_{i=1}^n \sum_{x \in \mathbb{F}_q^n : x_i = a} \mathrm{wt}_H(a)}{q^n}\\ &= \frac{q^{n-1} \sum_{i=1}^n \sum_{a \in \mathbb{F}_q} \mathrm{wt}_H(a)}{q^n}\\ &= \frac{q^{n-1} \sum_{i=1}^n (q-1)}{q^n}\\ &= n\frac{q-1}{q}.\end{aligned}$$

Where we have used that $q^{n-1}$ many $x \in \mathbb{F}_q^n$ are such that $x_i = a$ for a fixed $a \in \mathbb{F}_q$. □

In our previous example, this gives exactly the same.

**Example 5.6.** *Let us consider* $\overline{wt_H}(\mathbb{F}_3^3) = 3\frac{2}{3} = 2 = \overline{wt_H}(\mathcal{C})$.

This is always true.

**Lemma 5.7.** *Let* $\mathcal{C}$ *be a non-degenerate* $[n,k]_q$ *linear code. Then*

$$\overline{wt_H}(\mathcal{C}) = n\frac{q-1}{q}.$$

**Exercise 5.8.** *Repeat the proof of Lemma 5.5 for* $\mathcal{C}$*, thus proving Lemma 5.7.*

We can now put the two observations together, to get

$$d_H(\mathcal{C})(|\mathcal{C}|-1) \leq \sum_{c\in\mathcal{C}} \mathrm{wt}_H(c) = |\mathcal{C}|\overline{\mathrm{wt}_H}(\mathcal{C}) = |\mathcal{C}|n\frac{q-1}{q}$$

and dividing both sides by $|\mathcal{C}|-1$, we get the Plotkin bound.

**Theorem 5.9** (Plotkin Bound)**.** *Let* $q$ *be a prime power,* $k \leq n$ *be positive integers and* $\mathcal{C}$ *be a non-degenerate* $[n,k]_q$ *linear code. Then*

$$d_H(\mathcal{C}) \leq \frac{|\mathcal{C}|}{|\mathcal{C}|-1} n\frac{q-1}{q}.$$

While our previous argumentation is completely fine as a proof, we will also include a more standard proof here.

*Proof.* In this proof we construct a huge matrix, where the rows are given by all $c \in \mathcal{C}$. The resulting matrix $A \in \mathbb{F}_q^{q^k \times n}$. Using Lemma 5.7, we get that in each column every finite field element appears equally often, namely $q^{k-1}$ times. Since the number of zeroes in each column is $q^{k-1}$, we get that number of non-zero entries of $A$ is $nq^{k-1}(q-1)$. Similarly, going through the rows being $c \in \mathcal{C}$, we must have that the non-zero entries of $A$ are at least $d_H(\mathcal{C})(q^k-1)$, getting

$$d_H(\mathcal{C})(q^k-1) \leq nq^{k-1}(q-1).$$

This implies the claim. □

We can also prove the Plotkin bound using the Cauchy-Schwarz inequality.

**Lemma 5.10.** *Let* $n$ *be a positive integer and let* $x, y \in \mathbb{R}^n$*. Then*

$$\left(\sum_{i=1}^n x_i^2\right)\left(\sum_{i=1}^n y_i^2\right) \geq \left(\sum_{i=1}^n x_i y_i\right)^2.$$

In the following theorem, we do not assume that $\mathcal{C}$ is linear, hence we denote its cardinality simply by $M = |\mathcal{C}|$. As the code is not necessarily linear, we also have to work with $d_H(x, y)$ instead of $\mathrm{wt}_H(c)$.

**Theorem 5.11** (Plotkin Bound)**.** *Let $n$ be a positive integer and let $q$ be a prime power. Let $\mathcal{C} \subset \mathbb{F}_q^n$ with size $M > 2$ and minimum distance $d_H(\mathcal{C}) > n(q-1)/q$. Then*

$$|\mathcal{C}| \leq \frac{qd_H(\mathcal{C})}{qd_H(\mathcal{C}) - (q-1)n}.$$

*Proof.* We compute the sum

$$\sum_{x\in\mathcal{C}} \sum_{y\in\mathcal{C}} d_H(x, y)$$

in two different ways. We first observe that

$$\sum_{x\in\mathcal{C}} \sum_{y\in\mathcal{C}} d_H(x, y) \geq d_H(\mathcal{C})M(M-1),$$

by the definition of $d_H(\mathcal{C}) = \min\{d_H(x, y) \mid x \neq y \in \mathcal{C}\}$.

Let us denote by $\delta(a, b) = \begin{cases} 1 & \text{if } a \neq b, \\ 0 & \text{else.} \end{cases}$. With this we can write our sum as

$$\begin{aligned}
\sum_{x\in\mathcal{C}} \sum_{y\in\mathcal{C}} d_H(x, y) &= \sum_{i=1}^{n} \sum_{x\in\mathcal{C}} \sum_{y\in\mathcal{C}} \delta(x_i, y_i) \\
&= \sum_{i=1}^{n} \sum_{a\in\mathbb{F}_q} \sum_{x\in\mathcal{C}:x_i=a} \sum_{y\in\mathcal{C}} \delta(a, y_i) \\
&= \sum_{i=1}^{n} \sum_{a\in\mathbb{F}_q} |\{(x, y) \in \mathcal{C}^2 \mid x_i = a, y_i \neq a\}|.
\end{aligned}$$

Let us denote by $c(a, i) = |\{x \in \mathcal{C} \mid x_i = a\}|$, for some fixed $a \in \mathbb{F}_q$ and $i \in \{1, \ldots, n\}$. Then

$$\begin{aligned}
\sum_{x\in\mathcal{C}} \sum_{y\in\mathcal{C}} d_H(x, y) &= \sum_{i=1}^{n} \sum_{a\in\mathbb{F}_q} c(a, i)(M - c(a, i)) \\
&= nM^2 - \sum_{i=1}^{n} \sum_{a\in\mathbb{F}_q} c(a, i)^2.
\end{aligned}$$

We can now apply the Cauchy-Schwarz inequality to the vectors $x = (1, \ldots, 1)$ and $y = (c(a, i))_{a\in\mathbb{F}_q}$, that is

$$\sum_{a\in\mathbb{F}_q} 1^2 \sum_{a\in\mathbb{F}_q} c(a, i)^2 = q \sum_{a\in\mathbb{F}_q} c(a, i)^2 \geq \left( \sum_{a\in\mathbb{F}_q} c(a, i) \right)^2 = M^2,$$

as for all $x \in \mathcal{C}$ we must have some $a \in \mathbb{F}_q$ such that $x_i = a$.

Putting this inside our sum, we get

$$\begin{aligned}\sum_{x\in\mathcal{C}}\sum_{y\in\mathcal{C}} d_H(x,y) &= nM^2 - \sum_{i=1}^{n}\sum_{a\in\mathbb{F}_q} c(a,i)^2 \\ &\leq nM^2 - \sum_{i=1}^{n} M^2/q = nM^2 - nM^2q \\ &= nM^2\frac{q-1}{q}.\end{aligned}$$

Combining this with

$$\sum_{x\in\mathcal{C}}\sum_{y\in\mathcal{C}} d_H(x,y) \geq d_H(\mathcal{C})M(M-1),$$

we get that

$$d_H(\mathcal{C})M(M-1) \leq nM^2\frac{q-1}{q}$$

and thus dividing by $M$ on both sides, we get the claim. □

## 5.1 Simplex Code

We are again interested in codes which are optimal for this bound. As we have seen before

$$\sum_{c\in\mathcal{C}\setminus\{0\}} \mathrm{wt}_H(c) \geq (|\mathcal{C}|-1)d_H(\mathcal{C}),$$

only holds with equality if all non-zero codewords have minimal Hamming weight $d_H(\mathcal{C})$.

**Definition 5.12.** Let $1 \leq k \leq n$ be positive integers and $q$ be a prime power. Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. We say that $\mathcal{C}$ is a *constant weight code*, if for all $c \in \mathcal{C} \setminus \{0\}$ we have

$$\mathrm{wt}_H(c) = d_H(\mathcal{C}).$$

In case we are interested in non-linear codes, obtaining the non-linear version of the Plotkin bound, we would be asking for *equidistant* codes, i.e., for all $x \neq y \in \mathcal{C}$ we have

$$d_H(x,y) = d_H(\mathcal{C}).$$

We have already seen an optimal linear code for the Plotkin bound:

**Definition 5.13.** Let $q$ be a prime power and $r \geq 2$ be a positive integer. Let $n = \frac{q^r-1}{q-1}$ and $G$ be the $r \times n$ matrix having as columns all vectors in $\mathbb{F}_q^r$ up to non-zero scalar multiples. Then $\mathcal{C} = \langle G \rangle$ is called *simplex code.*

Recall that the simplex code $\mathcal{C}$ is a $[n, r]_q$ linear code. We now show that these codes are indeed optimal.

**Proposition 5.14.** *Let C be a simplex code then all non-zero codewords of $C$ have Hamming weight $q^{r-1}$.*

Thus if we plug in their minimum distance in the Plotkin bound, we get that

$$d_H(\mathcal{C}) = q^{r-1} = \frac{|\mathcal{C}|}{|\mathcal{C}| - 1} n \frac{q-1}{q} = \frac{q^r}{q^r - 1} \frac{q^r - 1}{q - 1} \frac{q-1}{q}.$$

*Proof.* By definition, a generator matrix $G$ of $\mathcal{C}$ has all vectors of $\mathbb{F}_q^r$ up to scalar multiples as columns. If we fix any non-zero codeword $c \in \mathcal{C}$, there exists a non-zero vector $m \in \mathbb{F}_q^r$ with $mG = c$.

We observe that for every non-zero $m \in \mathbb{F}_q^r$ there exist $\frac{q^{r-1}-1}{q-1}$ many non-zero vectors $h \in \mathbb{F}_q^r$ up to scalar multiples, such that $\langle m, h \rangle = 0$.

Thus, there are $\frac{q^{r-1}-1}{q-1}$ columns $h$ of $G$ such that $\langle m, h \rangle = 0$.

Hence the Hamming weight of $c = mG$ is given by

$$\frac{q^r - 1}{q - 1} - \frac{q^{r-1} - 1}{q - 1} = q^{r-1}.$$

□

**Example 5.15.** *Let us consider $q = 2$ and $r = 3$. Then a generator matrix of the simplex code is given by*

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

*We can check that all rows of $G$ have Hamming weight $d_H(\mathcal{C}) = q^{r-1} = 4$, and also any non-zero combination of the rows has weight 4.*

These are also essentially all optimal codes for the Plotkin bound. Indeed, according to the following result of Wood [20], we have that any constant weight code must be an $\ell$ fold of a simplex code.

**Theorem 5.16.** *Let $q$ be a prime power and $n$ a positive integer. Any constant weight code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is an $\ell$ fold of simplex codes.*

The proof of this theorem is unfortunately too complicated for a introductory course in coding theory. Instead we look at an example:

**Example 5.17.** *Let us consider $q = 2$ and $n = 6$. A two fold simplex code can be constructed by taking a generator matrix of a $[3, 2]_2$ simplex code*

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

*a permutation matrix $P \in \mathbb{F}_q^{n\times n}$ and considering the generator matrix*

$$G' = \begin{pmatrix} G & G \end{pmatrix} P.$$

*For example*

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

*which has constant weight* $4$.

**Exercise 5.18.** *Let $q$ be a prime power. For which lengths $n$ does there exist a constant weight code of length $n$?*

**Exercise 5.19.** *Let $q$ be a prime power. For which minimum distances $d$ does there exist a constant weight code of minimum distance $d$?*

## 5.2 Asymptotic Bound

Let us formulate the Plotkin bound in an asymptotic manner. For this, we again consider $k, d$ as a function in $n$ and set

$$\delta = \min_{n\to\infty} \frac{d(n)}{n}, \quad R = \limsup_{n\to\infty} \frac{k(n)}{n}$$

to be the relative minimum distance and the asymptotic information rate.

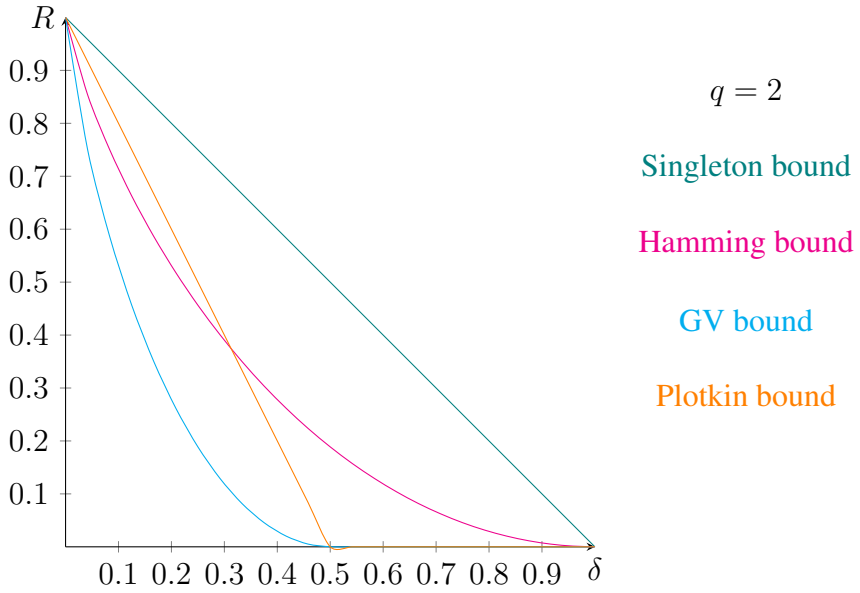
We can formulate the asymptotic Plotkin bound as the existence of a infinite family of codes as

**Theorem 5.20.** *Let $(\mathcal{C}_s)_s$ be a sequence of $[n_s, k_s, d_s]_q$ linear code. Then*

$$R \leq \max\{1 - \frac{q\delta}{q-1}, 0\}.$$

Unfortunately we do not have the right tools yet to prove this asymptotic formula, but we may come back to it later.

## 5.3 Comparison of the Bounds




We recall that the Gilbert-Varshamov bound is a lower bound (and an existence bound), while the Singleton, the Hamming and the Plotkin bound are all upper bounds on $d_H(\mathcal{C})$. Thus, we can see that the Singleton bound is by far the loosest lower bound, for rates $R < 0.3$, the Hamming bound provides a tighter lower bound than the Plotkin bound, but for a short range, roughly $R \in (0.35, 0.5)$ we have that the Plotkin bound is tighter.

## 5.4 Quick overview of other bounds

These are not all the bounds we have in coding theory, some other famous bounds are the Elias-Bassalygo bound and the Griesmer bound. We give here only a short overview of their statements.

The Griesmer bound is a lower bound, this time on $n$, when $k, d_H(\mathcal{C})$ are given.

**Theorem 5.21** (Griesmer Bound)**.** *Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. Then*

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil .$$

The proof requires residual codes, which we might encounter at a later point.

The Elias-Bassalygo bound can be seen as an upper bound on the size of the code $q^k$, when we fix its length $n$ and minimum distance $d_H(\mathcal{C})$.

**Theorem 5.22** (Elias-Bassalygo Bound)**.** *Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. Then*

$$q^k \leq qnd \frac{q^n}{|B_H(q,n,r,0)|},$$

*where* $r = \frac{q-1}{q}\left(n - \sqrt{n^2 - \frac{qnd}{q-1}}\right) - 1$.

To prove this, we first need to prove the Johnson bound. In fact, the value

$$J_q(\delta) = \frac{q-1}{q}\left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right)$$

is known as the Johnson radius.

- The *Plotkin bound* states $[n, k, d]_q$ linear codes are such that $d \leq \frac{q^k}{q^k-1} n \frac{q-1}{q}$.
- Codes attaining this bound are called *constant weight codes*.
- *Simplex* codes are constant weight codes.
- Any constant weight code is an $\ell$ fold of simplex codes.

# 6 Construction of New Codes

Given a $[n, k, d]_q$ linear code $\mathcal{C}$, there are many different ways to construct a new code $\mathcal{C}'$, with new parameters. In the following we will see how to extend, shorten, puncture, concatenate, add and multiply codes (and many more constructions).

## 6.1 Extension of Codes

We may recall the definition of a single parity-check code, which is

$$\mathcal{C} = \{c = (c_1, \ldots, c_k, -\sum_{i=1}^{k} c_i) \mid c_i \in \mathbb{F}_q\}.$$

The single parity-check code is a $[n, n-1, 2]_q$ linear code, generated by

$$G = \begin{pmatrix} & -1 \\ \mathrm{Id}_k & \vdots \\ & -1 \end{pmatrix}$$

and the dual code is the repetition code.

The encoding is thus simply given by $m = (m_1, \ldots, m_k) \in \mathbb{F}_q^k \mapsto c = (m_1, \ldots, m_k, -\sum_{i=1}^{k} m_i)$.

We may use a similar idea, to extend a code, instead of $m$.

**Definition 6.1.** Let $\mathcal{C}$ be a $[n, k, d]_q$ linear code. The *extended code* of $\mathcal{C}$ is defined as

$$\hat{\mathcal{C}} = \{(c_1, \ldots, c_n, -\sum_{i=1}^{n} c_i) \mid (c_1, \ldots, c_n) \in \mathcal{C}\}.$$

**Example 6.2.** *Let us consider the code*

$$\mathcal{C} = \{(0,0,0), (0,1,2), (0,2,1), (1,0,1), (1,1,0), (1,2,2), (2,0,2), (2,1,1), (2,2,0)\} \subset \mathbb{F}_3^3.$$

*Then,*

$$\begin{aligned}\hat{\mathcal{C}} = \{&(0,0,0,0),(0,1,2,0), (0,2,1,0), (1,0,1,1),\\ &(1,1,0,1), (1,2,2,1), (2,0,2,2), (2,1,1,2), (2,2,0,2)\}.\end{aligned}$$

The length of the new set is now $n+1$, but is it still a linear code, and what is the new dimension?

It turns out, that this extension does not change the linearity or the dimension.

**Proposition 6.3.** *Let $\mathcal{C}$ be a $[n, k, d]_q$ linear code. Then $\hat{\mathcal{C}}$ is a $[n+1, k, \geq d]_q$ linear code.*

*Proof.* Let $G \in \mathbb{F}_q^{k\times n}$ be a generator matrix of $\mathcal{C}$, i.e., for any $c \in \mathcal{C}$, there exists a $m \in \mathbb{F}_q^k$ such that $mG = c$, or equivalently for all $j \in \{1, \dots, n\}$ we have $c_j = \sum_{i=1}^k m_i g_{i,j}$.

We show that a generator matrix of the extended code is then given by

$$\hat{G} = \begin{pmatrix} & -\sum_{j=1}^n g_{1,j} \\ G & \vdots \\ & -\sum_{j=1}^n g_{k,j} \end{pmatrix}.$$

In fact, for any $m \in \mathbb{F}_q^k$ we have that

$$m\hat{G} = (\sum_{i=1}^k m_i g_{i,1}, \dots, \sum_{i=1}^k m_i g_{i,n}, -\sum_{j=1}^n \sum_{i=1}^k m_i g_{i,j}) \in \hat{\mathcal{C}},$$

thus $\langle \hat{G} \rangle \subseteq \hat{\mathcal{C}}$.

For the other direction, we observe that for every vector $\hat{c} = (c_1, \dots, c_n, -\sum_{j=1}^n c_j) \in \hat{\mathcal{C}}$, there exists a $m \in \mathbb{F}_q^k$ such that $\hat{c} = m\hat{G}$, thus $\hat{\mathcal{C}} \subseteq \langle \hat{G} \rangle$ and as $\mathrm{rk}(\hat{G}) = k$, we get that $\hat{\mathcal{C}}$ has length $n+1$ and dimension $k$.

Clearly, for all $c \in \mathcal{C}$ we have that $\mathrm{wt}_H(c) \leq \mathrm{wt}_H(\hat{c})$, hence we also get

$$d_H(\mathcal{C}) \leq d_H(\hat{\mathcal{C}}).$$

□

**Exercise 6.4.** *Given a parity-check matrix $H$ of $\mathcal{C}$, find a formulation for a parity-check matrix $\hat{H}$ of the extended code $\hat{\mathcal{C}}$.*

**Example 6.5.** *In our previous example, we see that $\mathcal{C} = \langle G \rangle \subset \mathbb{F}_3^3$, where*

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

*Thus, the extended code $\hat{\mathcal{C}}$ is generated by*

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

**Exercise 6.6.** *Let $\mathcal{C} \subset \mathbb{F}_2^n$. Show that, if $d_H(\mathcal{C})$ is odd, then $d_H(\hat{\mathcal{C}}) = d_H(\mathcal{C}) + 1$.*

If we can make codes longer, can we also make them shorter?

## 6.2 Puncturing

Clearly, we can just delete a position $i \in \{1, \dots, n\}$ over the whole code and obtain a shorter set. If we delete the last position, then we return form $\hat{\mathcal{C}}$ to $\mathcal{C}$.

However, we can also generalize this and *puncture* in any subset $S \subset \{1, \dots, n\}$.

**Definition 6.7.** Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code and $S \subset \{1, \dots, n\}$ be a subset of size $s$. The *punctured code* in $S$ is then

$$\mathcal{P}_S(\mathcal{C}) = \{(c_i)_{i \notin S} \mid c \in \mathcal{C}\}.$$

This is the opposite of the projection we have seen for the definition of the information set, i.e.,

$$\mathcal{C}_I = \{(c_i)_{i \in I} \mid c \in \mathcal{C}\},$$

i.e.,

$$\mathcal{P}_S(\mathcal{C}) = \mathcal{C}_{S^C}.$$

Clearly, the length of the new set is $n-s$, but is it still linear and what happens to its dimension?

**Proposition 6.8.** *Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code and $S \subset \{1, \dots, n\}$ be a subset of size $s < d$. Then $\mathcal{P}_S(\mathcal{C})$ is a $[n - s, k, \leq d]_q$ linear code.*

*Proof.* Let $G \in \mathbb{F}_q^{k \times n}$ be a generator matrix of $\mathcal{C}$. Since $\mathcal{P}_S(\mathcal{C}) = \mathcal{C}_{S^C}$, we recall that $G_{S^C}$, i.e., the columns of $G$ indexed by $S^C$ generate the code $\mathcal{P}_S(\mathcal{C})$.

However, the rank of $G_{S^C}$ might apriori drop. With the condition $s < d$, we ensure that this does not happen: for any two distinct codewords $c \neq c' \in \mathcal{C}$, we have that $d_H(c, c') \geq d$ and by deleting $s$ positions we get

$$d_H(c_{S^C}, c'_{S^C}) \geq d - s > 0,$$

hence they will not collide and we still have $q^k$ distinct codewords, which gives us that the dimension of $\mathcal{P}_S(\mathcal{C})$ is $k$.

Clearly, we must have $\mathrm{wt}_H(c) \geq \mathrm{wt}_H(c_{S^c})$ and hence $d_H(\mathcal{P}_S(\mathcal{C})) \leq d_H(\mathcal{C})$, but we can also bound it from below. Since we delete $s$ positions of $c$, we decrease its weight at most by $s$, i.e., $\mathrm{wt}_H(c_{S^C}) \geq \mathrm{wt}_H(c) - s$, with equality only if $S \subseteq \mathrm{supp}_H(c)$. □

**Example 6.9.** *Let us consider the code of our previous example again, namely $\hat{\mathcal{C}} = \langle \hat{G} \rangle \subset \mathbb{F}_3^4$, where*

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

*This code is a $[4, 2, 2]_3$ linear code. By puncturing in $S = \{4\}$ we recover the code $\mathcal{C}$, i.e.,*

$$\mathcal{P}_S(\hat{\mathcal{C}}) = \mathcal{C},$$

*which is a $[3, 2, 2]_3$ linear code.*

*If we puncture in $s \geq d = 2$, say $S' = \{2, 3\}$, we might decrease the dimension. In fact, in this case we get that*

$$\hat{G}_{S'^C} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},$$

*has rank 1 and thus we get that $\mathcal{P}_{S'}(\hat{\mathcal{C}})$ is a $[2, 1, 2]_3$ linear code.*

**Exercise 6.10.** *Is the extended RS code $\mathcal{RS}_{q,q+1,k}$ an extension of a code?*

## 6.3 Shortening

There also exists a different method to reduce the length, called shortening and it is in some sense (actually every sense) the dual of puncturing.

Shortening a code means to puncture a specific subcode of it.

**Definition 6.11.** Let $\mathcal{C}$ be a $[n, k, d]_q$ linear code and let $S \subset \{1, \ldots, n\}$ be a subset of size $s$. Let us consider the subcode

$$\mathcal{C}(S) = \{c \in \mathcal{C} \mid c_i = 0 \; \forall i \in S\}.$$

Then the *shortened code* in $S$ is given by

$$\mathcal{S}_S(\mathcal{C}) = \mathcal{P}_S(\mathcal{C}(S)) = \mathcal{C}(S)_{S^C}.$$

Since any punctured code is still linear, so is $\mathcal{C}(S)$. How will a parity-check matrix for the shortened code look like?

**Lemma 6.12.** *Let $\mathcal{C}$ be a $[n, k, d]_q$ linear code and let $S \subset \{1, \ldots, n\}$ be a subset of size $s$. Let $c' \in \mathcal{S}_S(\mathcal{C})$, then there exists a codeword $c \in \mathcal{C}$ such that*

$$c_i = \begin{cases} c'_i & \text{if } i \notin S, \\ 0 & \text{if } i \in S \end{cases}$$

*for all $i \in \{1, \ldots, n\}$.*

**Exercise 6.13.** *Prove Lemma 6.12.*

**Lemma 6.14.** *Let $\mathcal{C}$ be a $[n, k, d]_q$ linear code and let $S \subset \{1, \ldots, n\}$ be a subset of size $s$. Let $H \in \mathbb{F}_q^{(n-k)\times n}$ be a parity-check matrix of $\mathcal{C}$ and define $H' = H_{S^C}$ to be the matrix obtained by deleting all columns of $H$ indexed by $S$. Then $\mathcal{S}_S(\mathcal{C}) = ker(H'^\top)$.*

*Proof.* Since $c \in \mathcal{C}$ we have that $cH^\top = 0$ and hence for any $c' \in \mathcal{S}_S(\mathcal{C})$ we get that $c'H'^\top = 0$, which implies that $\mathcal{S}_S(\mathcal{C}) \subset \ker(H'^\top)$.

For the other direction, let us consider a vector $c' \in \mathbb{F}_q^{n-s}$ with $c'H^\top = 0$. We can now construct a $c \in \mathcal{C}$ using Lemma 6.12, i.e.,

$$c_i = \begin{cases} c'_i & \text{if } i \notin S, \\ 0 & \text{if } i \in S \end{cases}$$

for all $i \in \{1, \ldots, n\}$.

Since $c'H'^\top = 0$, we must also get that $cH^\top = 0$ and hence $c \in \mathcal{C}$, and in turn $c' \in \mathcal{S}_S(\mathcal{C})$. □

**Example 6.15.** *Let us again consider* $\hat{\mathcal{C}} = \langle \hat{G} \rangle \subset \mathbb{F}_3^4$, *where*

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

*Hence* $\hat{G}$ *is a parity-check matrix of* $\hat{\mathcal{C}}^\perp$ *and to get a parity-check matrix* $H'$ *of* $\mathcal{S}_{S'}(\hat{\mathcal{C}}^\perp)$, *for* $S' = \{2, 3\}$ *we delete the second and third column, getting*

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

This parity-check matrix reminds us of how we constructed the generator matrix for the punctured code. In fact, these two notions are dual.

**Theorem 6.16.** *Let* $\mathcal{C}$ *be a* $[n, k, d]_q$ *linear code and let* $S \subset \{1, \ldots, n\}$ *be a subset of size* $s$.

1. $\mathcal{S}_S(\mathcal{C}^\perp) = (\mathcal{P}_S(\mathcal{C}))^\perp$, *and*
2. $\mathcal{P}_S(\mathcal{C}^\perp) = (\mathcal{S}_S(\mathcal{C}))^\perp$.

*Proof.* Let $G \in \mathbb{F}_q^{k \times n}$ be a generator matrix of $\mathcal{C}$ and $H \in \mathbb{F}_q^{(n-k) \times n}$ be a parity-check matrix. Recall that $G$ is then a parity-check matrix of $\mathcal{C}^\perp$ and $H$ is a generator matrix of $\mathcal{C}^\perp$.

We have seen that $G_{S^C}$ is then a generator matrix for $\mathcal{P}_S(\mathcal{C})$ and note that $G_{S^C}$ is also how we constructed the parity-check matrix of a shortened code, that is we get the first claim.

For the second claim we can argue similarly, as $\mathcal{S}_S(\mathcal{C}) = \ker(H_{S^C}^\top)$ and thus, $H_{S^C}$ is a generator matrix of $\mathcal{S}_S(\mathcal{C})^\perp$. Since $H$ is a generator matrix of $\mathcal{C}^\perp$, by the same construction as before, we get that $H_{S^C}$ is a generator matrix of $\mathcal{P}_S(\mathcal{C}^\perp)$, implying the second claim. □

Now we can quickly deduce what happens to the code parameters.

**Corollary 6.17.** *Let* $\mathcal{C}$ *be a* $[n, k, d]_q$ *linear code and let* $S \subset \{1, \ldots, n\}$ *be a subset of size* $s < d_H(\mathcal{C}^\perp)$. *Then* $\mathcal{S}_S(\mathcal{C})$ *is a* $[n - s, k - s, \geq d]_q$ *linear code.*

The corollary follows directly from the parameters of the punctured code and Theorem 6.16, as

$$\mathcal{S}_S(\mathcal{C})^\perp = \mathcal{P}_S(\mathcal{C}^\perp),$$

thus if $s < d_H(\mathcal{C}^\perp)$, then $\mathcal{P}_S(\mathcal{C}^\perp)$ has parameters $[n - s, n - k, \leq d_H(\mathcal{C}^\perp)]_q$ and thus, $\mathcal{S}_S(\mathcal{C})$ has parameters $[n - s, k - s, \geq d]_q$.

**Example 6.18.** *Let us again consider* $\hat{\mathcal{C}} = \langle \hat{G} \rangle \subset \mathbb{F}_3^4$, *where*

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

*We have seen that puncturing $\hat{\mathcal{C}}$ in $S' = \{2, 3\}$ gives us $\mathcal{P}'_S(\hat{\mathcal{C}}) = \langle G' \rangle$, where*

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

*This means that*

$$\mathcal{P}_{S'}(\hat{\mathcal{C}}) = \{(0,0), (1,1), (2,2)\} \subset \mathbb{F}_3^2.$$

*Since $\hat{G}$ is a parity-check matrix of $\hat{\mathcal{C}}^\perp$ we get a parity-check matrix $H'$ of $\mathcal{S}_{S'}(\hat{\mathcal{C}}^\perp)$ as*

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

*Clearly,*

$$\langle G' \rangle = \mathcal{P}_{S'}(\hat{\mathcal{C}}) = \langle H' \rangle = (\mathcal{S}_{S'}(\hat{\mathcal{C}}^\perp))^\perp.$$

**Example 6.19.** *Let $\alpha \in \mathbb{F}_q^n$ be such that $\alpha_i \neq \alpha_j$ for all $i \neq j$ and $\beta \in (\mathbb{F}_q^\star)^n$. Let $S \subset \{1, \ldots, n\}$ of size $s < n - k + 1$, then*

$$\mathcal{P}_S(\mathcal{GRS}_{q,n,k}(\alpha, \beta)) = \mathcal{GRS}_{q,n-s,k}(\alpha', \beta'),$$

*where*

$$\alpha' = (\alpha_i)_{i \notin S}, \quad \beta' = (\beta_i)_{i \notin S}.$$

*By duality of shortening and puncturing, also the shortened GRS code is a GRS code:*

$$\begin{aligned} \mathcal{S}_S(\mathcal{GRS}_{q,n,k}(\alpha, \beta)^\perp) &= \mathcal{S}_S(\mathcal{GRS}_{q,n,n-k}(\alpha, \gamma)) = \mathcal{P}_S(\mathcal{GRS}_{q,n,k}(\alpha, \beta))^\perp \\ &= \mathcal{GRS}_{q,n-s,k}(\alpha', \beta'))^\perp = \mathcal{GRS}_{q,n-s,n-s-k}(\alpha', \gamma')), \end{aligned}$$

*where*

$$\gamma_i = \beta_i^{-1} \prod_{j=1, j \neq i}^{n} (\alpha_i - \alpha_j)^{-1}, \quad \gamma' = (\gamma_i)_{i \notin S}.$$

## 6.4 Product of codes

The easiest way to think of combining two codes $\mathcal{C}_1, \mathcal{C}_2$ is to simply put the codewords of $\mathcal{C}_2$ after those of $\mathcal{C}_1$ :

**Definition 6.20.** Let $n_1, n_2$ be two positive integers and $k_1 \leq n_1, k_2 \leq n_2$ be positive integers. Let $\mathcal{C}_1$ be a $[n_1, k_1]_q$ linear code and $\mathcal{C}_2$ be a $[n_2, k_2]_q$ linear code. The *product code* of $\mathcal{C}_1$ and $\mathcal{C}_2$ is given by

$$\mathcal{C}_1 \times \mathcal{C}_2 = \{(c_1, c_2) \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\}.$$

This construction is also called direct sum of $\mathcal{C}_1, \mathcal{C}_2$.
Clearly, the new length is now $n_1 + n_2$. What happens to the other parameters?

**Proposition 6.21.** *Let $k_1, d_1 \leq n_1$ and $k_2, d_2 \leq n_2$ be positive integers. Let $\mathcal{C}_1$ be a $[n_1, k_1, d_1]_q$ linear code and $\mathcal{C}_2$ be a $[n_2, k_2, d_2]_q$ linear code. The product code $\mathcal{C}_1 \times \mathcal{C}_2$ is a $[n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\}]_q$ linear code.*

*Proof.* Let $G_1$ be a generator matrix of $\mathcal{C}_1$ and $G_2$ a generator matrix of $\mathcal{C}_2$. We first show that $\mathcal{C}_1 \times \mathcal{C}_2$ is linear, by showing that

$$G = \begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix}$$

is a generator matrix of $\mathcal{C}_1 \times \mathcal{C}_2$.

For any codeword $c = (c_1, c_2) \in \mathcal{C}_1 \times \mathcal{C}_2$, there exists a $m = (m_1, m_2) \in \mathbb{F}_q^{k_1+k_2}$ such that $c = mG$.

Indeed, since $c_1 \in \mathcal{C}_1$, there exists a $m_1 \in \mathbb{F}_q^{k_1}$ such that $c_1 = m_1 G_1$ and similarly, since $c_2 \in \mathcal{C}_2$, there exists a $m_2 \in \mathbb{F}_q^{k_2}$ such that $c_2 = m_2 G_2$, thus $\langle G \rangle = \mathcal{C}_1 \times \mathcal{C}_2$.

Since $\text{rk}(G_1) = k_1$ and $\text{rk}(G_2) = k_2$, we immediately get that $\text{rk}(G) = k_1 + k_2$.

Finally, for the minimum distance, we note that the minimal weight codeword $c_1$ of $\mathcal{C}_1$ is such that $(c_1, 0) \in \mathcal{C}_1 \times \mathcal{C}_2$ with $\text{wt}_H((c_1, 0)) = d_1$ and similarly, the minimal weight codeword $c_2$ of $\mathcal{C}_2$ is such that $(0, c_2) \in \mathcal{C}_1 \times \mathcal{C}_2$ with $\text{wt}_H((0, c_2)) = d_2$.

Since any codeword $(x, y)$ has weight $\geq d_1$ and in the same way $\geq d_2$, we get that

$$d_H(\mathcal{C}_1 \times \mathcal{C}_2) = \min\{d_1, d_2\}.$$

□

**Example 6.22.** *let $\mathcal{C}_1$ be a $[3, 2, 2]_3$ linear code generated by*

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

*and $\mathcal{C}_2$ be a $[2, 1, 2]_3$ linear code generated by $G_2 = \begin{pmatrix} 1 & 2 \end{pmatrix}$.*

*Their product code $\mathcal{C}_1 \times \mathcal{C}_2$ is then generated by*

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

## 6.5 Plotkin Sum

This construction is also called $(u, u + v)$ construction, which is a huge spoiler.

**Definition 6.23.** Let $n$ be a positive integer and $k_1, k_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2]_q$ linear code. The *Plotkin sum* of $\mathcal{C}_1$ and $\mathcal{C}_2$ is given by

$$\mathcal{C}_1 +_P \mathcal{C}_2 = \{(c_1, c_1 + c_2) \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\}.$$

The length of the new set is again clear: $2n$.

**Proposition 6.24.** *Let $n$ be a positive integer and $k_1, k_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2]_q$ linear code. The Plotkin sum $\mathcal{C}_1 +_P \mathcal{C}_2$ is a $[2n, k_1 + k_2]_q$ linear code.*

**Exercise 6.25.** *Prove Proposition 6.24 by writing the generator matrix of $\mathcal{C}_1 +_P \mathcal{C}_2$ in terms of the generator matrix of $\mathcal{C}_1$ and the generator matrix of $\mathcal{C}_2$.*

**Exercise 6.26.** *Let $n$ be a positive integer and $k_1, k_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2]_q$ linear code.*

*Is it true that $\mathcal{C}_1 +_P \mathcal{C}_2 = \mathcal{C}_2 +_P \mathcal{C}_1$?*

**Example 6.27.** *Let us consider the repetition code $\mathcal{C}$, which is a $[n, 1, n]_q$ linear code. Then*

$$\mathcal{C} +_P \mathcal{C} = \langle \begin{pmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ 0 & \cdots & 0 & 1 & \cdots & 1 \end{pmatrix} \rangle$$

*is a $[2n, 2, n]_q$ linear code.*

An important property of such Plotkin sum codes is the following.

**Proposition 6.28.** *Let $n$ be a positive integer and $k_1, k_2, d_1, d_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1, d_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2, d_2]_q$ linear code. The Plotkin sum $\mathcal{C}_1 +_P \mathcal{C}_2$ has minimum Hamming distance min$\{2d_1, d_2\}$.*

*Proof.* Let $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$ be the minimum weight codewords. Clearly, $(c_1, c_1+0) \in \mathcal{C}_1+_P\mathcal{C}_2$ and $\text{wt}_H((c_1, c_1)) = 2d_1$. We also have that $(0, c_2) \in \mathcal{C}_1 +_P \mathcal{C}_2$ and thus

$$d_H(\mathcal{C}_1 +_P \mathcal{C}_2) \leq \min\{2d_1, d_2\}.$$

Let us consider now any $c = (x, x + y) \in \mathcal{C}_1 +_P \mathcal{C}_2$, which is non-zero. If $y = 0$, then $x \neq 0$ and $\text{wt}_H(c) \geq 2d_1$.

If $y \neq 0$, then to get a non-zero codeword $c$, we must have that if $y_i \neq 0$, then either $x_i \neq 0$ or $y_i \neq -x_i$. Thus, the weight of $c$ is at least $\text{wt}_H(y) \geq d_2$. □

If we know a decoding algorithm $\text{Dec}_1$ for $\mathcal{C}_1$ and a decoding algorithm $\text{Dec}_2$ for $\mathcal{C}_2$, we can then devise a decoding strategy for their Plotkin sum:

Assume we received the vector $r = (r_1, r_2) \in \mathbb{F}_q^{2n}$.

1. Use $\text{Dec}_1$ to decode $r_1$ and recover the codeword $c_1 \in \mathcal{C}_1$.
2. Compute $r_2' = r_2 - c_1 = c_2 + e$ and decode $r_2'$ using $\text{Dec}_2$ to recover $c_2$.
3. Recover the sent codeword $c = (c_1, c_1 + c_2)$.

## 6.6 Sum of Codes

The usual definition of a sum of sets would be the following construction.

**Definition 6.29.** Let $n$ be a positive integer and $k_1, k_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2]_q$ linear code. The *sum* of $\mathcal{C}_1$ and $\mathcal{C}_2$ is given by

$$\mathcal{C}_1 + \mathcal{C}_2 = \{c_1 + c_2 \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\}.$$

**Exercise 6.30.** *Is it true that $\mathcal{C}_1 +_P \mathcal{C}_2 = \mathcal{C}_1 \times (\mathcal{C}_1 + \mathcal{C}_2)$?*

**Proposition 6.31.** *Let $n$ be a positive integer and $k_1, k_2, d_1, d_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1, d_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2, d_2]_q$ linear code. The sum $\mathcal{C}_1 + \mathcal{C}_2$ is a $[n, \leq k_1 + k_2, \leq \min\{d_1, d_2\}]_q$ linear code.*

*Proof.* We first show that $\mathcal{C}_1 + \mathcal{C}_2$ is generated by $G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}$, where $\mathcal{C}_1 = \langle G_1 \rangle, \mathcal{C}_2 = \langle G_2 \rangle$.

Clearly, for any codeword $c = (c_1 + c_2)$ there exists a message vector $m = (m_1 + m_2) \in \mathbb{F}_q^{k_1+k_2}$ such that $c = mG$, as we can choose $m_1, m_2$ such that $c_1 = m_1 G$ and $c_2 = m_2 G$.

However, the dimension of $\mathcal{C}_1 + \mathcal{C}_2$ is then the rank of $G$, which might not be $k_1 + k_2$, in fact, think of the case $G_1 = G_2$.

If $\text{rk}(G) = k_1 + k_2 - \ell$, then there are $\ell$ rows of $G_2$ which are linearly dependent to the rows of $G_1$, and their linear combinations give $q^\ell$ codewords which live in both $\mathcal{C}_1$ and $\mathcal{C}_2$.

Hence,

$$\dim(\mathcal{C}_1 + \mathcal{C}_2) = \text{rk}(G) - \ell = \text{rk}(G) - \dim(\mathcal{C}_1 \cap \mathcal{C}_2).$$

The minimum distance is harder to control. We know however, that $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}_1 + \mathcal{C}_2$ as subcodes and hence $d_H(\mathcal{C}_1 + \mathcal{C}_2) \leq \min\{d_1, d_2\}$. □

**Example 6.32.** *Let us consider $\mathbb{F}_5$ and the two RS codes generated by*

$$G_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 4 & 1 \\ 4 & 4 & 1 & 1 \end{pmatrix}.$$

*Then $\mathcal{C}_1$ is a $[4, 2, 3]_5$ linear code and $\mathcal{C}_2$ is a $[4, 3, 2]_5$ linear code. The sum $\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$ is a $[4, 4, 1]_5$ linear code and thus $\mathcal{C} = \mathbb{F}_5^4$.*

## 6.7 Intersection of Codes

Given two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ we can also consider their intersection, i.e., codewords which live in both codes.

**Definition 6.33.** Let $n$ be a positive integer and $k_1, k_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2]_q$ linear code. The *intersection* of $\mathcal{C}_1$ and $\mathcal{C}_2$ is given by

$$\mathcal{C}_1 \cap \mathcal{C}_2 = \{c \mid c \in \mathcal{C}_1, c \in \mathcal{C}_2\}.$$

**Proposition 6.34.** *Let $n$ be a positive integer and $k_1, k_2, d_1, d_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1, d_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2, d_2]_q$ linear code. The intersection $\mathcal{C}_1 \cap \mathcal{C}_2$ is a $[n, \leq \min\{k_1, k_2\}, \geq \max\{d_1, d_2\}]_q$ linear code.*

*Proof.* We show that $H = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix}$ has $\mathcal{C}_1 \cap \mathcal{C}_2$ as kernel, for $\mathcal{C}_1 = \ker(H_1^\top)$ and $\mathcal{C}_2 = \ker(H_2^\top)$.

This is again clear, as any $c \in \mathcal{C}_1 \cap \mathcal{C}_2$ must be such that $cH_1^\top = 0$ and $cH_2^\top = 0$.

Since $\mathcal{C}_1 \cap \mathcal{C}_2$ is a subcode of both $\mathcal{C}_1$ and $\mathcal{C}_2$ we easily get the dimension $\leq \min\{k_1, k_2\}$ and the minimum distance $d_H(\mathcal{C}_1 \cap \mathcal{C}_2) \geq \max\{d_1, d_2\}$.

However, we can also say more: the dimension of $\mathcal{C}_1 \cap \mathcal{C}_2$ is given by the rank-nullity theorem as

$$\dim(\mathcal{C}_1 \cap \mathcal{C}_2) = n - \mathrm{rk}(H) = n - (2n - (k_1 + k_2) - \ell').$$

As before we have that $\ell'$ denotes the rank deficiency of $H$, implying that there are codewords living in both $\langle H_1 \rangle = \mathcal{C}_1^\perp$ and $\langle H_2 \rangle = \mathcal{C}_2^\perp$.

Thus,

$$\dim(\mathcal{C}_1 \cap \mathcal{C}_2) = (k_1 + k_2) - n + \dim(\mathcal{C}_1^\perp \cap \mathcal{C}_2^\perp).$$

□

We can see a clear connection to the sum of $\mathcal{C}_1$ and $\mathcal{C}_2$. In fact, these two constructions are dual to each other:

**Proposition 6.35.** *Let $n$ be a positive integer and $k_1, k_2 \leq n$ be positive integers. Let $\mathcal{C}_1$ be a $[n, k_1]_q$ linear code and $\mathcal{C}_2$ be a $[n, k_2]_q$ linear code. Then $(\mathcal{C}_1 \cap \mathcal{C}_2)^\perp = \mathcal{C}_1^\perp + \mathcal{C}_2^\perp$.*

*Proof.* Let $\mathcal{C}_1 = \ker(H_1)^\top$ and $\mathcal{C}_2 = \ker(H_2)^\top$. We have seen that $\mathcal{C}_1 \cap \mathcal{C}_2 = \ker\left(\begin{pmatrix} H_1 \\ H_2 \end{pmatrix}^\top\right)$ and hence

$$(\mathcal{C}_1 \cap \mathcal{C}_2)^\perp = \left\langle \begin{pmatrix} H_1 \\ H_2 \end{pmatrix} \right\rangle = \mathcal{C}_1^\perp + \mathcal{C}_2^\perp.$$

□

Note that the hull of a $[n, k]_q$ linear code $\mathcal{C}$, i.e., $\mathcal{H}(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^\perp$ is also a code intersection. Thus, we can write the dual of the hull as a sum:

$$\mathcal{H}(\mathcal{C})^\perp = \mathcal{C}^\perp + \mathcal{C}.$$

## 6.8 Expansion Codes

If a code $\mathcal{C}$ is $\mathbb{F}_{q^m}$-linear, for some prime power $q$ and positive integer $m$, it is also $\mathbb{F}_q$-linear. Thus, we can also think of it as a code in $\mathbb{F}_q^{mn}$.

For this, let us recall the expansion map.

**Definition 6.36.** Let $\Gamma = \{\gamma_0, \ldots, \gamma_{m-1}\}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then the expansion map with respect to $\Gamma$ is given by

$$\begin{aligned} \exp_\Gamma : \mathbb{F}_{q^m} &\to \mathbb{F}_q^m, \\ a = \sum_{i=0}^{m-1} a_i\gamma_i &\mapsto \exp_\Gamma(a) = (a_0, \ldots, a_{m-1}). \end{aligned}$$

**Definition 6.37.** Let $q$ be a prime power, $m$ a positive integer and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k]_{q^m}$ linear code. The *expanded code* of $\mathcal{C}$ is

$$\Gamma(\mathcal{C}) = \{\exp_\Gamma(c) \in \mathbb{F}_q^n \mid c \in \mathcal{C}\}.$$

By abuse of notation, we will introduce writing $\Gamma(x)$ instead of $\exp_\Gamma(x)$ for any $x \in \mathbb{F}_{q^m}^n$. Clearly, the expanded code depends on the choice basis $\Gamma$.

**Example 6.38.** *Let us consider $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ where $\alpha^2 = \alpha + 1$.*

*Let us consider the code $\mathcal{C}$ generated by*

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

*that is*

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

*Let us choose the polynomial basis $\Gamma = \{1, \alpha\}$. Then*

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

*If we however choose the basis $\Gamma' = \{\alpha, 1\}$ we get*

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

*We can easily check that these are not the same codes, e.g.* $(1,0,0,0,0,1) \in \Gamma(\mathcal{C})$ *but* $(1,0,0,0,0,1) \notin \Gamma'(\mathcal{C})$

**Proposition 6.39.** *Let* $q$ *be a prime power,* $m$ *a positive integer and* $k, d \leq n$ *be positive integers. Let* $\mathcal{C}$ *be an* $[n,k,d]_{q^m}$ *linear code. The expanded code* $\Gamma(\mathcal{C})$ *is a* $[mn, mk, \leq dm]_q$ *linear code.*

*Proof.* Recall that $\exp_\Gamma$ is an isomorphism between $\mathbb{F}_q$-vector spaces. Hence we keep the $\mathbb{F}_q$ linearity, getting a linear code and since

$$|\mathcal{C}| = |\Gamma(\mathcal{C})| = q^{mk}.$$

With this we immediately get length $mn$ and dimension $mk$.

Clearly, for any $a \in \mathbb{F}_{q^m}$ we have that $\mathrm{wt}_H(a) \leq \mathrm{wt}_H(\Gamma(a)) \leq m\mathrm{wt}_H(a)$. □

As in the previous example the minimum distance stayed the same, let us also show an example where it increases.

**Example 6.40.** *Let us consider* $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ *where* $\alpha^2 = \alpha + 1$.

*Let us consider the code* $\mathcal{C}$ *generated by*

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

*that is*

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

*We have* $d_H(\mathcal{C}) = 3$.

*Let us choose the polynomial basis* $\Gamma = \{1, \alpha\}$. *Then*

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

*In this case we get* $d_H(\Gamma(\mathcal{C})) = 4$.

We can also write down a generator matrix for $\Gamma(\mathcal{C})$ in terms of the generator matrix of $\mathcal{C}$.

**Proposition 6.41.** *Let $\Gamma = \{\gamma_0, \ldots, \gamma_{m-1}\}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Let $\mathcal{C}$ be a $[n,k]_{q^m}$ linear code with generator matrix $G = \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix}$, where $g_i \in \mathbb{F}_{q^m}^n$ are the rows of $G$.*

*Then a generator matrix $\Gamma(G)$ of $\Gamma(\mathcal{C})$ is given by*

$$\Gamma(G) = \begin{pmatrix} \Gamma(g_1\gamma_0) \\ \vdots \\ \Gamma(g_1\gamma_{m-1}) \\ \vdots \\ \Gamma(g_k\gamma_0) \\ \vdots \\ \Gamma(g_k\gamma_{m-1}) \end{pmatrix} \in \mathbb{F}_q^{mk \times mn}.$$

That is, every entry $g_{i,j}$ of $G$ is expanded to its multiplication matrix $M_\Gamma(g_{i,j})$.

*Proof.* Let us take a random codeword $c \in \mathcal{C}$, then there exists a $m \in \mathbb{F}_{q^m}^k$ such that $c = mG$. Since $\Gamma(c) = (\Gamma(c_1), \ldots, \Gamma(c_n))$, it will be enough to consider $\Gamma(c_i)$. We want to show that

$$\Gamma(c_i) = \Gamma(m) \begin{pmatrix} M_\Gamma(g_{1,i}) \\ \vdots \\ M_\Gamma(g_{k,i}) \end{pmatrix},$$

which implies $\Gamma(\mathcal{C}) \subseteq \langle \Gamma(G) \rangle$.

From $c = mG$, we get that $c_i = \sum_{j=1}^k m_j g_{j,i}$ for all $i \in \{1, \ldots, n\}$. Thus

$$\begin{aligned} \Gamma(c_i) &= \Gamma(\sum_{j=1}^k m_j g_{j,i}) = \sum_{j=1}^k \Gamma(m_j g_{j,i}) \\ &= \sum_{j=1}^k \Gamma(m_j) M_\Gamma(g_{j,i}) = \Gamma(m) \begin{pmatrix} M_\Gamma(g_{1,i}) \\ \vdots \\ M_\Gamma(g_{k,i}) \end{pmatrix}, \end{aligned}$$

where we have used that to keep linearity, we have to involve the multiplication matrix, that is: if $a, b \in \mathbb{F}_{q^m}$ we have $\Gamma(ab) = \Gamma(a)M_\Gamma(b)$.

At the same time, we recall that $\Gamma(\mathcal{C})$ has dimension $mk$, thus it will be enough to show that $\Gamma(G)$ has full rank $km$.

We can see this by considering $G$ in systematic form, i.e., there exists an information set $I$, with $G_I = \mathrm{Id}_k$. Let us denote by $e_i$ the standard vector of length $k$ having all entries zero except

for the $i$th entry being 1. Then,

$$\Gamma(G_I) = \begin{pmatrix} \Gamma(e_1\gamma_0) \\ \vdots \\ \Gamma(e_1\gamma_{m-1}) \\ \vdots \\ \Gamma(e_k\gamma_0) \\ \vdots \\ \Gamma(e_k\gamma_{m-1}) \end{pmatrix} = \begin{pmatrix} e'_1 \\ \vdots \\ e'_m \\ \vdots \\ e'_{mk-m} \\ \vdots \\ e'_{mk} \end{pmatrix},$$

where $e'_i$ is the standard vector of length $mk$.

Thus, $\mathrm{rk}(\Gamma(G)) = mk$ and hence $\Gamma(\mathcal{C}) = \langle \Gamma(G) \rangle$.

□

**Exercise 6.42.** *Write down a parity-check matrix, called $\Gamma(H)$, of $\Gamma(\mathcal{C})$ in terms of the parity-check matrix $H$ of $\mathcal{C}$.*

**Proposition 6.43.** *Let $q$ be a prime power, $m$ a positive integer, $\Gamma = \{\gamma_0, \ldots, \gamma_{m-1}\}$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k]_{q^m}$ linear code with generator matrix $G \in \mathbb{F}_{q^m}^{k \times n}$. Then $\Gamma(mG) = \Gamma(m)\Gamma(G)$ for all $m \in \mathbb{F}_{q^m}^k$.*

*Proof.* Let $m = (m_1, \ldots, m_k) \in \mathbb{F}_{q^m}^k$ and $m_i = \sum_{j=0}^{m-1} m_{i,j}\gamma_j$, and $\Gamma(G)$ as in the proof of Proposition 6.47, then

$$\begin{aligned} \Gamma(m)\Gamma(G) &= \sum_{i=1}^{k} \sum_{j=0}^{m-1} m_{i,j}\Gamma(g_i\gamma_j) \\ &= \sum_{i=1}^{k} \Gamma(\sum_{j=0}^{m-1} m_{i,j} g_i \gamma_j) \\ &= \sum_{i=1}^{k} \Gamma(m_i g_i) \\ &= \Gamma(\sum_{i=1}^{k} m_i g_i) = \Gamma(mG). \end{aligned}$$

□

**Exercise 6.44.** *Let $H$ be a parity-check matrix of $\mathcal{C}$ and $\Gamma(H)$ a parity-check matrix of $\Gamma(\mathcal{C})$. Show that $\Gamma(Hy^\top) = \Gamma(H)\Gamma(y)^\top$ for all $y \in \mathbb{F}_{q^m}^n$.*

## 6.9 Subfield Subcodes

Given a linear code $\mathcal{C} \subset \mathbb{F}_{q^m}$ for some prime power $q$ and positive integer $m$, we can also consider the subcode living only in a subfield $\mathbb{F}_q$.

**Definition 6.45.** Let $q$ be a prime power, $m$ a positive integer and $k \leq n$ be positive integers. Let $\mathcal{C}$ be an $[n,k]_{q^m}$ linear code. The *subfield subcode* of $\mathcal{C}$ is

$$\mathcal{C}_{\mathbb{F}_q} = \{c \in \mathbb{F}_q^n \mid c \in \mathcal{C}\} = \mathcal{C} \cap \mathbb{F}_q^n.$$

By Proposition 6.35 we have that

$$\mathcal{C}_{\mathbb{F}_q} = (\mathcal{C} \cap \mathbb{F}_q^n)^\perp = \mathcal{C}^\perp + (\mathbb{F}_q^n)^\perp = \mathcal{C}^\perp + \{0\} = \mathcal{C}^\perp.$$

This is true thinking of $\mathcal{C}_{\mathbb{F}_q} \subset \mathbb{F}_{q^m}^n$, i.e., as a $\mathbb{F}_{q^m}$-linear subspace. However, when considering subfield subcodes, we are more interested in seeing them as $\mathbb{F}_q$-linear subspace and in this case we have that

$$\mathcal{C}_{\mathbb{F}_q}^\perp = \mathcal{C}^\perp \cap \mathbb{F}_q^n.$$

With this special subcode construction, we still have the same length, but in general we should reduce the dimension (recall that $\mathcal{C}$, seen as $\mathbb{F}_q$-linear code, has dimension $mk$), and due to the fact $\mathcal{C}_{\mathbb{F}_q} \subset \mathcal{C}$ we have that $d_H(\mathcal{C}_{\mathbb{F}_q}) \geq d_H(\mathcal{C})$. For the dimension, we can say even more:

**Proposition 6.46.** *Let $n$ be a positive integer and $k, d \leq n$ be positive integers. Let $\mathcal{C}$ be a $[n,k,d]_q$ linear code. The subfield subcode $\mathcal{C}_{\mathbb{F}_q}$ is a $[n, \geq km - n(m-1), \geq d]_q$ linear code.*

*Proof.* Let us consider the map

$$\begin{aligned} \phi : \mathbb{F}_{q^m} &\to \mathbb{F}_{q^m}, \\ x &\mapsto x^q - x. \end{aligned}$$

We clearly get that $\mathbb{F}_q = \ker(\phi)$. We can extend this componentwise to consider

$$\begin{aligned} \phi : \mathbb{F}_{q^m}^n &\to \mathbb{F}_{q^m}^n, \\ (x_1, \ldots, x_n) &\mapsto (x_1^q - x_1, \ldots, x_n^q - x_n). \end{aligned}$$

Hence $\dim(\ker(\phi)) = n$ and thus, $\dim(\mathrm{im}(\phi)) = nm - n$. If we restrict $\phi$ to $\mathcal{C}$, i.e., $\phi_{|\mathcal{C}} : \mathcal{C} \to \mathbb{F}_{q^m}^n$ we get that $\ker(\phi_{|\mathcal{C}})$ are exactly the codewords which live in $\mathbb{F}_q^n$, i.e., $\mathcal{C}_{\mathbb{F}_q}$. We also get that $\mathrm{im}(\phi_{|\mathcal{C}})$ has $\mathbb{F}_{q^m}$- dimension at most $nm - n$, thus

$$\dim_{\mathbb{F}_q}(\ker(\phi_{|\mathcal{C}})) \geq \dim_{\mathbb{F}_q}(\mathcal{C}) - nm + n = mk - nm + n.$$

□

If we consider again the expanded code of $\mathcal{C}$, we can see a connection between $\Gamma(\mathcal{C})$ and $\mathcal{C}_{\mathbb{F}_q}$:

**Proposition 6.47.** *Let $\mathcal{C}$ be a $[n,k]_{q^m}$ linear code. Let $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of $\mathcal{C}$, then*

$$\mathcal{C}_{\mathbb{F}_q} = \textit{ker}\left(\begin{pmatrix}\Gamma(h_1)^\top & \cdots & \Gamma(h_n)^\top\end{pmatrix}\right).$$

Thus, a subfield subcode is a shortened expanded code $\Gamma(\mathcal{C})$. This is now also independent on the choice of basis $\Gamma$.

*Proof.* By definition the subfield subcode of $\ker(H^\top)$ is the $\mathbb{F}_q$-kernel of $H^\top$, that is any $c \in \mathbb{F}_q^n$ such that $Hc^\top = 0$.

Let $c \in \ker_{\mathbb{F}_q}(H^\top)$, that is $c \in \mathcal{C}_{\mathbb{F}_q}$, then

$$\sum_{i=1}^{n} h_{j,i}c_i = 0$$

for all $j \in \{1,\dots,n-k\}$.

Fix the basis $\Gamma = \{1, a, \dots, a^{m-1}\}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then we can write

$$h_{j,i} = \sum_{\ell=0}^{m-1} a^\ell h_{j,i,\ell},$$

with $h_{j,i,\ell} \in \mathbb{F}_q$.

Then

$$\Gamma(\sum_{i=1}^{n} h_{j,i}c_i) = \Gamma(0) = (0,\dots,0)$$

and since $c_i \in \mathbb{F}_q$, we get

$$\Gamma(\sum_{i=1}^{n} h_{j,i}c_i) = \sum_{i=1}^{n}\Gamma(h_{j,i})c_i = (\sum_{i=1}^{n} h_{j,i,0}c_i, \dots, \sum_{i=1}^{n} h_{j,i,m-1}c_i) = (0,\dots,0)$$

for all $j \in \{1,\dots,n-k\}$.

Thus,

$$\begin{pmatrix}\Gamma(h_1)^\top & \cdots & \Gamma(h_n)^\top\end{pmatrix} c^\top = \begin{pmatrix} h_{1,1,0} & & h_{1,n,0} \\ \vdots & & \vdots \\ h_{1,1,m-1} & & h_{1,n,m-1} \\ \vdots & \cdots & \vdots \\ h_{n-k,1,0} & & h_{n-k,n,0} \\ \vdots & & \vdots \\ h_{n-k,1,m-1} & & h_{n-k,n,m-1} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = 0.$$

The other direction works in the same way. □

### 6.9.1 Alternant Codes

We have seen that GRS codes are optimal in terms of their error-correction capability. Unfortunately, they are limited in their length by $n \leq q+1$. And so over the binary, no non-trivial MDS codes exist. However, we can consider subfield subcodes of MDS codes, getting *alternant codes.*

**Definition 6.48.** Let $\alpha \in \mathbb{F}_{q^m}^n$ have pairwise distinct entries, $\beta \in (\mathbb{F}_{q^m}^\star)^n$. The *alternant code* is then defined as

$$\mathcal{A}_{q,m,n,k}(\alpha,\beta) = (\mathcal{GRS}_{q^m,n,k}(\alpha,\beta)_{\mathbb{F}_q}.$$

**Corollary 6.49.** *The alternant code $\mathcal{A}_{q,m,n,k}(\alpha,\beta)$ is a $[n, \geq n-m(n-k), \geq n-k+1]_q$ linear code.*

The dimension directly follows from being a subfield subcode of a $[n,k]_{q^m}$ linear code:

$$\dim(\mathcal{A}_{q,m,n,k}(\alpha,\beta)) \geq km - n(m-1),$$

and the minimum distance of a subfield subcode is at least the minimum distance of the original code, in our case $d_H(\mathcal{GRS}_{q^m,n,k}(\alpha,\beta)) = n-k+1$. Alternant codes are not optimal, we get as rate

$$R' \geq \frac{km-nm+n}{n} = 1 - m + \log_q(n)R,$$

where $R = \frac{k}{n}$ is the rate of the original GRS code, and their relative minimum distance stays the same

$$\delta' \geq \delta = \frac{d}{n} = \frac{n-k+1}{n} \sim 1 - R.$$

Thus, we are far away from being optimal with respect to the Singleton bound $\delta' \leq 1 - R'$. However, the minimum distance is usually much larger than $d$.

**Example 6.50.** *Let us consider $\mathbb{F}_9 = \mathbb{F}_3(\alpha)$ where $\alpha^2 = \alpha + 1$. Let us consider the $[3,2,2]_9$ RS code $\mathcal{C}$ generated by*

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

*and with parity-check matrix*

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

*Let $\Gamma = \{1, \alpha\}$, then*

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

*Note that ker$(\Gamma(H)^\top) = \Gamma(\mathcal{C})$. The subfield subcode $\mathcal{C}_{\mathbb{F}_3}$ is the kernel of $\phi_{|\mathcal{C}} \to \mathbb{F}_9^3$ and consists only of $\mathcal{C}_{\mathbb{F}_3} = \{(0,0,0),(1,1,1),(2,2,2)\}$, which is a $[3,1,3]_3$ linear code.*

*The same code is generated by taking the parity-check matrix*

$$\begin{pmatrix} \Gamma(h_1)^\top & \Gamma(h_2)^\top & \Gamma(h_3)^\top \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \end{pmatrix}.$$

### 6.9.2 Goppa Codes

One special subclass of alternant codes are *Goppa codes* introduced by Goppa [5]. There are several different definitions for this class of codes, in particular there exist the algebraic geometry version of the Goppa codes and the so-called "classical" Goppa codes. We will stick to the classical version.

**Proposition 6.51.** *Let $g(x) \in \mathbb{F}_{q^m}[x]$ be a polynomial of degree $s \geq 1$. Let $\alpha \in \mathbb{F}_{q^m}^n$ consist of pairwise distinct entries. If $g(\alpha_i) \neq 0$ for all $i \in \{1, \ldots, n\}$, then there exist $h_i(x) \in \mathbb{F}_{q^m}[x]$ with*

$$(x - \alpha_i)h_i(x) \equiv 1 \mod g(x).$$

*In fact,*

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

**Exercise 6.52.** *Prove Proposition 6.51.*

**Example 6.53.** *Let us consider $\mathbb{F}_4 = \mathbb{F}_2(a)$ with $a^2 = a + 1$ and $g(x) = 1 + ax + x^2$. We may choose $\alpha = (0, 1, a, a + 1)$ since $g(x)$ is irreducible.*

*Using Proposition 6.51, we get*

$$\begin{aligned}
h_1(x) &= \frac{1 + ax + x^2 + 1}{x} 1^{-1} = a + x, \\
h_2(x) &= \frac{1 + ax + x^2 + a}{x + 1} a^{-1} = a + (1 + a)x, \\
h_3(x) &= \frac{1 + ax + x^2 + 1}{x + a} 1^{-1} = x, \\
h_4(x) &= \frac{1 + ax + x^2 + a}{x + (1 + a)} a^{-1} = (1 + a) + (1 + a)x.
\end{aligned}$$

**Definition 6.54.** Let $g(x) \in \mathbb{F}_{q^m}[x]$ be a polynomial of degree $s \geq 1$. Let $\alpha \in \mathbb{F}_{q^m}^n$ consist of pairwise distinct entries, such that $g(\alpha_i) \neq 0$ for all $i \in \{1, \ldots, n\}$. The *Goppa code* with *Goppa polynomial* $g(x)$ is then defined as

$$\Gamma_{q,m,n}(g(x), \alpha) = \{c \in \mathbb{F}_q^n \mid \sum_{i=1}^{n} c_i h_i(x) \equiv 0 \mod g(x)\}.$$

**Example 6.55.** *Let us consider again $\mathbb{F}_4 = \mathbb{F}_2(a)$ with $a^2 = a + 1$ and $g(x) = 1 + ax + x^2$. We may choose $\alpha = (0, 1, a, a + 1)$ with $h_1(x) = a + x, h_2(x) = a + (1 + a)x, h_3(x) = x$ and $h_4(x) = (1 + a) + (1 + a)x$.*

*The only $c \in \mathbb{F}_2^4$ which is such that $\sum_{i=1}^n c_i h_i(x) \equiv 0 \mod g(x)$ is $c = (0, 0, 0, 0)$.*

*Let us thus consider $g(x) = 1+ax$, with $\alpha = (0, 1, a)$ and $h_1(x) = a, h_2(x) = 1+a, h_3(x) = 1$. Thus, $\Gamma_{2,2,3}(1 + ax, (0, 1, a)) = \{(0, 0, 0), (1, 1, 1)\}$.*

The drop in the degree of $g(x)$ resulted in a larger dimension.

**Proposition 6.56.** *Let $g(x) \in \mathbb{F}_{q^m}[x]$ be a polynomial of degree $s \geq 1$. Let $\alpha \in \mathbb{F}_{q^m}^n$ consist of pairwise distinct entries with $g(\alpha_i) \neq 0$ for all $i \in \{1, \ldots, n\}$. The alternant code $\mathcal{A}_{q,m,n,n-s}(\alpha, \beta)$, for*

$$\beta_i = g(\alpha_i) \prod_{i \neq j} (\alpha_i - \alpha_j)^{-1}$$

*is exactly the Goppa code $\Gamma_{q,m,n}(g(x), \alpha)$.*

*Proof.* First, let us denote by $g(\alpha)^{-1} = (g(\alpha_1)^{-1}, \ldots, g(\alpha_n)^{-1})$.

The Goppa code consists of all the $c \in \mathbb{F}_q^n$ such that $\sum_{i=1}^n c_i h_i(x) \equiv 0 \mod g(x)$. Since $\deg(h_i) < s$, this implies

$$\sum_{i=1}^n c_i h_i(x) = -\sum_{i=1}^n g(\alpha_i)^{-1} c_i \frac{g(x) - g(\alpha_i)}{x - \alpha_i} = 0$$

is the zero polynomial.

We may write out the polynomial

$$\begin{aligned} \frac{g(x) - g(\alpha_i)}{x - \alpha_i} &= g_s(x^{s-1} + x^{s-2}\alpha_i + \cdots + \alpha_i^{s-1}) \\ &\quad + g_{s-1}(x^{s-2} + \cdots + \alpha_i^{s-2}) + g_2(x + \alpha_i) + g_1. \end{aligned}$$

By setting the coefficients of $x^i$ to 0 in $\sum_{i=1}^n c_i h_i(x)$, we hence get that $c \in \Gamma_{q,m,n}(g(x), \alpha)$ if and only if $Hc^\top = 0$, where

$$\begin{aligned} H &= \begin{pmatrix} g_s & \cdots & g_s \\ g_{s-1} + \alpha_1 g_s & \cdots & g_{s-1} + \alpha_n g_s \\ \vdots & & \vdots \\ g_1 + \alpha_1 g_2 + \cdots + \alpha_1^{s-1} g_s & \cdots & g_1 + \alpha_n g_2 + \cdots + \alpha_n^{s-1} g_s \end{pmatrix} \operatorname{diag}(g(\alpha)^{-1}) \\ &= \begin{pmatrix} g_s & 0 & \cdots & 0 \\ g_{s-1} & g_s & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ g_1 & g_2 & \cdots & g_s \end{pmatrix} \begin{pmatrix} 1 & \cdots & 1 \\ \alpha_1 & \cdots & \alpha_n \\ \vdots & & \vdots \\ \alpha_1^{s-1} & \cdots & \alpha_n^{s-1} \end{pmatrix} \operatorname{diag}(g(\alpha)^{-1}) = SV_{q^m,n,s}(\alpha, g(\alpha)^{-1}). \end{aligned}$$

Since $S$ is an invertible matrix, we get that $H$ generates the same code as

$$\begin{pmatrix} 1 & \cdots & 1 \\ \alpha_1 & \cdots & \alpha_n \\ \vdots & & \vdots \\ \alpha_1^{s-1} & \cdots & \alpha_n^{s-1} \end{pmatrix} \begin{pmatrix} g(\alpha_1)^{-1} & & \\ & \ddots & \\ & & g(\alpha_n)^{-1} \end{pmatrix} \in \mathbb{F}_{q^m}^{s \times n}.$$

We can now easily check that for $\gamma = g(\alpha)^{-1}$ we get

$$\beta_i = g(\alpha_i) \prod_{i \neq j} (\alpha_i - \alpha_j)^{-1}$$

and thus the claim.

□

Since the Goppa code is a subfield subcode we can immediately bound its dimension and minimum distance.

**Corollary 6.57.** *Let $g(x) \in \mathbb{F}_{q^m}[x]$ be a polynomial of degree $s \geq 1$. Let $\alpha \in \mathbb{F}_{q^m}^n$ consist of pairwise distinct entries with $g(\alpha_i) \neq 0$ for all $i \in \{1, \ldots, n\}$. The $\Gamma_{q,m,n}(g(x), \alpha)$ Goppa code is a $[n, \geq n - ms, \geq s + 1]_q$ linear code.*

## 6.10 Trace Codes

There exists also another way to go from an extension field to a subfield and keeping the linearity, i.e., by using the trace. For this we recall the trace function:

**Definition 6.58.** Let $\mathbb{F}_{q^m}$ and $\mathbb{F}_q$ be the finite field with $q^m$, respectively $q$, elements. The *trace* map is defined as

$$\begin{aligned} \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q} : \mathbb{F}_{q^m} &\to \mathbb{F}_q, \\ \alpha &\mapsto \sum_{i=0}^{m-1} \alpha^{q^i}. \end{aligned}$$

For convenience (and when there is no ambiguity to which subfield we are going) we will simply write Tr instead of $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$.

**Definition 6.59.** Let $\mathcal{C}$ be an $[n, k]_{q^m}$ linear code. The *trace code* is then defined as

$$\mathrm{Tr}(\mathcal{C}) = \{(\mathrm{Tr}(c_1), \ldots, \mathrm{Tr}(c_n)) \mid (c_1, \ldots, c_n) \in \mathcal{C}\}.$$

Since $\mathrm{Tr} : \mathbb{F}_{q^m} \to \mathbb{F}_q$ the new set now lives in $\mathbb{F}_q^n$.

**Proposition 6.60.** *Let $\mathcal{C}$ be an $[n, k, d]_{q^m}$ linear code. The trace code $Tr(\mathcal{C})$ is then a $[n, \leq mk, \leq d]_q$ linear code.*

*Proof.* Recall that $\dim_{\mathbb{F}_q}(\mathcal{C}) = km$ and Tr is a $\mathbb{F}_q$-linear map, thus

$$\mathrm{Tr}_{|\mathcal{C}} : \mathcal{C} \to \mathbb{F}_q^n$$

has $\mathrm{im}(\mathrm{Tr}_{|\mathcal{C}}) = \mathrm{Tr}(\mathcal{C})$ and is of dimension $\leq mk$.

For the minimum distance, we observe that $\mathrm{Tr}(0) = 0$ and thus, $\mathrm{wt}_H(\mathrm{Tr}(c)) \leq \mathrm{wt}_H(c)$. □

How can we express the generator matrix and the parity-check matrix of $\mathrm{Tr}(\mathcal{C})$ in terms of $G, H$ the generator, respectively the parity-check matrix of $\mathcal{C}$? For this we first need to introduce the *dual basis*.

**Proposition 6.61.** *Let $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then there exists a unique basis $\Gamma' = \{\gamma'_1, \ldots, \gamma'_m\}$ such that*

$$Tr(\gamma_i \gamma'_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else.} \end{cases}$$

*This $\Gamma'$ is called* dual basis *to $\Gamma$.*

The uniqueness and existence follow from the fact that given a non-degenerate bilinear map, any basis has a dual basis.

**Corollary 6.62.** *Let $\Gamma$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ and dual basis $\Gamma'$. Let $x \in \mathbb{F}_{q^m}$. Then we may write*

$$x = \sum_{i=1}^{m} Tr(\gamma'_i x)\gamma_i.$$

This follows directly as the decomposition $x = \sum_{i=1}^{m} x_i \gamma_i$ with $x_i \in \mathbb{F}_q$ is unique. Thus, for any $i \in \{1, \ldots, m\}$ we have that

$$\mathrm{Tr}(\gamma'_i x) = \sum_{j=1}^{m} \mathrm{Tr}(\gamma'_i \gamma_j) x_j = x_i.$$

Similarly, for a vector $x \in \mathbb{F}_{q^m}^n$, we may write

$$x = \sum_{i=1}^{m} \gamma_i \mathrm{Tr}(\gamma'_i x).$$

**Proposition 6.63.** *Let $\mathcal{C}$ be an $[n, k]_{q^m}$ linear code and $\Gamma$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ with dual basis $\Gamma'$. Let $G = \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix}$ be a generator matrix of $\mathcal{C}$. Then $Tr(\mathcal{C})$ is generated by*

$$\begin{pmatrix} Tr(\gamma'_1 g_1) \\ \vdots \\ Tr(\gamma'_m g_1) \\ \vdots \\ Tr(\gamma'_1 g_k) \\ \vdots \\ Tr(\gamma'_m g_k) \end{pmatrix}.$$

**Exercise 6.64.** *Prove Proposition 6.63.*

**Example 6.65.** *Let us consider* $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ *with* $\alpha^2 = \alpha + 1$. *Let* $\mathcal{C} = \langle G \rangle$ *where*

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

*then Tr*$(\mathcal{C})$ *is generated by*

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

The trace code and the subfield subcode are also dual to each other! First, we have to differentiate whether we take the inner product in $\mathbb{F}_q$, i.e., $\langle \cdot, \cdot \rangle_{\mathbb{F}_q}$ or in $\mathbb{F}_{q^m}$, i.e., $\langle \cdot, \cdot \rangle_{\mathbb{F}_{q^m}}$.

**Theorem 6.66** (Delsarte Theorem)**.** *Let* $\mathcal{C}$ *be an* $[n,k]_{q^m}$ *linear code. Then* $(\mathcal{C}_{\mathbb{F}_q})^\perp =$ *Tr*$(\mathcal{C}^\perp)$.

*Proof.* Let $u \in \mathcal{C}^\perp$ and $v \in \mathcal{C}_{\mathbb{F}_q}$, then

$$\langle \mathrm{Tr}(u), v \rangle_{\mathbb{F}_q} = \sum_{i=1}^{n} \mathrm{Tr}(u_i) v_i = \sum_{i=1}^{n} \mathrm{Tr}(u_i v_i) = \mathrm{Tr}(\langle u, v \rangle_{\mathbb{F}_{q^m}}) = 0,$$

where we have used that $v_i \in \mathbb{F}_q$. Hence we get the first direction: $\mathrm{Tr}(\mathcal{C}^\perp) \subseteq (\mathcal{C}_{\mathbb{F}_q})^\perp$.

For the other direction, let us consider $u \in (\mathrm{Tr}(\mathcal{C}^\perp))^\perp$ and $v \in \mathcal{C}^\perp, \lambda \in \mathbb{F}_{q^m}$. Then

$$\mathrm{Tr}(\lambda \langle u, v \rangle_{\mathbb{F}_{q^m}}) = \mathrm{Tr}(\lambda \sum_{i=1}^{n} u_i v_i) = \sum_{i=1}^{n} \mathrm{Tr}(\lambda u_i v_i).$$

Since $u \in (\mathrm{Tr}(\mathcal{C}^\perp))^\perp$ is in $\mathbb{F}_q^n$, we get

$$\mathrm{Tr}(\lambda \langle u, v \rangle_{\mathbb{F}_{q^m}}) = \sum_{i=1}^{n} u_i \mathrm{Tr}(\lambda v_i) = \langle u, \mathrm{Tr}(\lambda v) \rangle_{\mathbb{F}_{q^m}}.$$

Since $\mathcal{C}^\perp$ is $\mathbb{F}_{q^m}$ linear, we have $\lambda v \in \mathcal{C}^\perp$ and thus for all $\lambda \in \mathbb{F}_{q^m}$ we get

$$\mathrm{Tr}(\lambda \langle u, v \rangle_{\mathbb{F}_{q^m}}) = 0.$$

Thus $\langle u, v \rangle_{\mathbb{F}_{q^m}} = 0$, as $\mathrm{Tr}(\lambda x) = 0$ for all $\lambda$ implies $x = 0$.

Hence $(\mathrm{Tr}(\mathcal{C}^\perp))^\perp \subseteq (\mathcal{C}^\perp)^\perp = \mathcal{C}$. As $(\mathrm{Tr}(\mathcal{C}^\perp))^\perp \subset \mathbb{F}_q^n$, we also get

$$(\mathrm{Tr}(\mathcal{C}^\perp))^\perp \subset \mathcal{C} \cap \mathbb{F}_q^n = \mathcal{C}_{\mathbb{F}_q}.$$

□

## 6.11 Power Codes

One construction has been used several times to distinguish structured codes (e.g. RS codes) from random codes, breaking cryptosystems trying to hide this code structure. This construction is called square code, or in more generality, power code.

**Definition 6.67.** Let $x, y \in \mathbb{F}_q^n$. Let us denote by $*$ the componentwise product or *Schur product* between $x$ and $y$, i.e.,

$$x * y = (x_1 y_1, \ldots, x_n y_n).$$

The Schur product has many nice properties, in particular, it is symmetric and bilinear. That means

1. For $\lambda \in \mathbb{F}_q$ we have $(\lambda x) * y = x * (\lambda y) = \lambda(x * y)$.
2. For $z \in \mathbb{F}_q^n$ we have $(x + z) * y = x * y + z * y$ and $x * (y + z) = x * y + x * z$.
3. We have $x * y = x * y$.

We can thus also consider this operation on codes, leading to

**Definition 6.68.** Let $\mathcal{C}_1$ be an $[n, k_1]_q$ linear code and $\mathcal{C}_2$ be an $[n, k_2]_q$ linear code. The *Schur product code* of $\mathcal{C}_1$ and $\mathcal{C}_2$ is defined as

$$\mathcal{C}_1 * \mathcal{C}_2 = \langle \{c_1 * c_2 \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\} \rangle.$$

If we apply this to the same code, we get the *square code* of $\mathcal{C}$, defined as

$$\mathcal{C}^{(2)} = \langle \{c * c' \mid c, c' \in \mathcal{C}\} \rangle.$$

**Example 6.69.** *Let us consider the* $[3, 2]_3$ *linear code* $\mathcal{C}$ *generated by* $G = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix}$. *That is*

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

*Then the square code is given by*

$$\mathcal{C}^{(2)} = \mathbb{F}_3^3.$$

**Proposition 6.70.** *Let* $\mathcal{C}$ *be generated by* $G = \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix} \in \mathbb{F}_q^{k \times n}$. *Then* $\mathcal{C}^{(2)}$ *is generated by*

$$G^{(2)} = \begin{pmatrix} g_1 * g_1 \\ \vdots \\ g_1 * g_k \\ \vdots \\ g_k * g_k \end{pmatrix} \in \mathbb{F}_q^{\binom{k+1}{2} \times n}.$$

*Proof.* Let $c \in \mathcal{C}^{(2)}$, then there exist $c_1, c_2 \in \mathcal{C}$ such that $c = c_1 * c_2$. Hence we have $m_1, m_2 \in \mathbb{F}_q^k$ such that $c_1 = m_1 G = \sum_{i=1}^k m_{1,i} g_i$ and $c_2 = m_2 G = \sum_{i=1}^k m_{2,i} g_i$.

Thus,

$$\begin{aligned} c = m_1 G * m_2 G &= (\sum_{i=1}^k m_{1,i} g_{i,1} \cdot \sum_{i=1}^k m_{2,i} g_{i,1}, \ldots, \sum_{i=1}^k m_{1,i} g_{i,n} \cdot \sum_{i=1}^k m_{2,i} g_{i,n}) \\ &= (\sum_{i,j=1}^k (g_{i,1} g_{j_1})(m_{1,i} m_{2,j}), \ldots, \sum_{i,j=1}^k (g_{i,n} g_{j_n})(m_{1,i} m_{2,j})) \\ &= M G^{(2)}, \end{aligned}$$

where $M = (m_{1,1}m_{2,1}, m_{1,1}m_{2,2}, \ldots, m_{1,k}m_{2,k})$. □

**Proposition 6.71.** *Let $g_1, \ldots, g_k \in \mathbb{F}_q^n$ be linearly independent over $\mathbb{F}_q$. Then for $x \in (\mathbb{F}_q^\star)^n$, also $x * g_1, \ldots, x * g_k$ are linearly independent.*

**Exercise 6.72.** *Prove Proposition 6.71.*

Using this fact, it was shown that with high probability (for increasing $n$), random codes have square codes of maximal dimension:

**Theorem 6.73.** *Let $\mathcal{C}$ be a random $[n,k]_q$ linear code. Then with high probability*

$$\dim(\mathcal{C}^{(2)}) = \min\left\{ \binom{k+1}{2}, n \right\}.$$

However, for more structured code the square code dimension is much lower.

**Proposition 6.74.** *Let $\alpha \in \mathbb{F}_q^n$ be such that $\alpha_i \neq \alpha_j$ for $i \neq j$ and $\beta, \beta' \in (\mathbb{F}_q^\star)^n$. Let $k' = \min\{n, 2k-1\}$. Then*

$$\mathcal{GRS}_{q,n,k}(\alpha, \beta) * \mathcal{GRS}_{q,n,k}(\alpha, \beta') = \mathcal{GRS}_{q,n,k'}(\alpha, \beta * \beta').$$

**Exercise 6.75.** *Prove Proposition 6.74.*

Thus,

$$\dim(\mathcal{GRS}_{q,n,k}(\alpha, \beta)^{(2)}) = \min\{n, 2k-1\} \leq \min\left\{ \binom{k+1}{2}, n \right\},$$

which is the expected dimension of a square code of a random code.

**Exercise 6.76.** *Let $\mathcal{C}$ be self-orthogonal. Show that $(1, \ldots, 1) \in (\mathcal{C}^{(2)})^\perp$.*

We can also generalize this idea and take larger powers.

**Definition 6.77.** Let $\mathcal{C}$ be a $[n,k]_q$ linear code. The $\ell$th *power code* of $\mathcal{C}$ is defined as

$$\mathcal{C}^{(\ell)} = \langle \{ \underbrace{c_1 * \cdots * c_\ell}_{\ell \text{ times}} \mid c_1, \ldots, c_\ell \in \mathcal{C} \} \rangle.$$

We can construct again a generator matrix taking as rows all $\underbrace{g_1 * \cdots * g_\ell}_{\ell \text{ times}}$, where $g_i$ are rows of $G$. The dimension of a $\ell$th power code of a random code is now a bit more complicated, but has been shown to be with high probability

$$\dim(\mathcal{C}^{(\ell)}) = \binom{k+\ell-1}{\ell},$$

for $\ell \in \{0, \ldots, q\}$.

**Exercise 6.78.** *Find a description of $\mathcal{GRS}_{q,n,k}(\alpha,\beta)^{(\ell)}$, for $\ell \in \{0,\ldots,q\}$.*

## 6.12 Concatenation

As last construction, we want to consider concatenation of codes. This construction was introduced by Forney [4] and is encoding a message vector twice:

**Definition 6.79.** Let $\mathcal{C}_1$ be a $[n_1,k_1,d_1]_q$ linear code, called *inner code* and $\mathcal{C}_2$ be an $[n_2,k_2,d_2]_{q^{k_1}}$ linear code, called *outer code*. The *concatenated code* $\mathcal{C}_2 \circ \mathcal{C}_1$ is then defined through the following encoding: Let $\Gamma$ be a basis of $\mathbb{F}_{q^{k_1}}$ over $\mathbb{F}_q$

$$\begin{aligned}
&\mathbb{F}_q^{k_1k_2} \overset{\exp_\Gamma^{-1}}{\to} \mathbb{F}_{q^{k_1}}^{k_2} \overset{\mathcal{C}_2}{\to} \mathbb{F}_{q^{k_1}}^{n_1} \overset{\exp_\Gamma}{\to} \mathbb{F}_q^{k_1n_2} \overset{\mathcal{C}_1}{\to} \mathbb{F}_q^{n_1n_2}\\
&(\exp_\Gamma(u_1),\ldots,\exp_\Gamma(u_{k_2})) \mapsto (u_1,\ldots,u_{k_2}) \mapsto ((uG_2)_1,\ldots,(uG_2)_{n_2})\\
&\quad\mapsto (\exp_\Gamma((uG_2)_1),\ldots,\exp_\Gamma((uG_2)_{n_2})) \mapsto (\exp_\Gamma((uG_2)_1)G_1,\ldots,\exp_\Gamma((uG_2)_{n_2})G_1).
\end{aligned}$$

**Proposition 6.80.** *Let $\mathcal{C}_1$ be a $[n_1,k_1,d_1]_q$ linear code and $\mathcal{C}_2$ be an $[n_2,k_2,d_2]_{q^{k_1}}$ linear code. The* concatenated code $\mathcal{C}_2 \circ \mathcal{C}_1$ *is a $[n_1n_2, k_1k_2, \geq d_1d_2]_q$ linear code.*

**Exercise 6.81.** *Prove Proposition 6.80.*

**Example 6.82.** *Let us consider $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ where $\alpha^2 = \alpha + 1$ and the basis $\Gamma = \{1,\alpha\}$. Let $\mathcal{C}_1$ be a $[4,2]_2$ linear code generated by $G_1 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$ and $\mathcal{C}_2$ be a $[3,2]_4$ linear code generated by $G_2 = \begin{pmatrix} 1 & 0 & \alpha \\ 0 & 1 & \alpha+1 \end{pmatrix}$.*

*The concatenated code $\mathcal{C}_2 \circ \mathcal{C}_1$ is then the encoding map*

$$\mathbb{F}_2^4 \to \mathbb{F}_4^2 \to \mathbb{F}_4^3 \to \mathbb{F}_2^6 \to \mathbb{F}_2^{12}.$$

*If we for example want to encode the message $(1,0,0,1)$ then we first compute $exp_\Gamma^{-1}(1,0) = 1, exp_\Gamma^{-1}(0,1) = \alpha$. Thus, we encode $(1,\alpha)$ using $G_2$ getting $(1,\alpha,\alpha+1)$. We can then expand the outer codeword to get $(1,0,0,1,1,1)$ and encode each expanded vector using $G_1$ to get*

$$(1,0,1,0,0,1,1,1,1,1,0,1).$$

The concatenated code can be thought of as a subcode of $\underbrace{\mathcal{C}_1 \times \cdots \times \mathcal{C}_1}_{n_2 \text{ times}}$, as the resulting codeword lives in $\mathcal{C}_1 \times \cdots \times \mathcal{C}_1$, but we are not able to see all possible codewords, only those, which are codewords of $\exp_\Gamma(\mathcal{C}_2)$.

**Exercise 6.83.** *Can you find an outer code $\mathcal{C}_2$, such that $\mathcal{C}_2 \circ \mathcal{C}_1 = \mathcal{C}_1$?*

In order to decode, we have to reverse the chain, i.e., we first apply the decoder of the inner code $\mathcal{C}_1$, then we apply $\exp_\Gamma^{-1}$, then the decoder of $\mathcal{C}_2$ and finally, we apply $\exp_\Gamma$. For simplicity, let us denote the decoding of $\mathcal{C}_i$ by $\mathcal{C}_i^{-1}$, then we get the decoding process as

$$\mathbb{F}_q^{n_1 n_2} \overset{\mathcal{C}_1^{-1}}{\to} \mathbb{F}_q^{k_1 n_2} \overset{\exp_\Gamma^{-1}}{\to} \mathbb{F}_{q^{k_1}}^{n_1} \overset{\mathcal{C}_2^{-1}}{\to} \mathbb{F}_{q^{k_1}}^{k_2} \overset{\exp_\Gamma}{\to} \mathbb{F}_q^{k_1 k_2}.$$

Summary

| Construction | $\mathcal{C}_1$ | $\mathcal{C}_2$ | $\to \mathcal{C}$ |
|---|---|---|---|
| Extension | $[n,k,d]_q$ | | $[n+1,k \geq d]_q$ |
| Puncturing | $[n,k,d]_q$ | | $[n-s,k,\leq d]_q$ |
| Shortening | $[n,k,d]_q$ | | $[n-s,k-s,\geq d]_q$ |
| Product | $[n_1,k_1,d_1]_q$ | $[n_2,k_2,d_2]_q$ | $[n_1+n_2,k_1+k_2,\min\{d_1,d_2\}]_q$ |
| Plotkin sum | $[n,k_1,d_1]_q$ | $[n,k_2,d_2]_q$ | $[2n,k_1+k_2,\min\{2d_1,d_2\}]_q$ |
| Sum | $[n,k_1,d_1]_q$ | $[n,k_2,d_2]_q$ | $[n,\leq k_1+k_2,\leq \min\{d_1,d_2\}]_q$ |
| Intersection | $[n,k_1,d_1]_q$ | $[n,k_2,d_2]_q$ | $[n,\leq \min\{k_1,k_2\},\geq \max\{d_1,d_2\}]_q$ |
| Expansion | $[n,k,d]_{q^m}$ | | $[mn,mk,\leq dm]_q$ |
| Subfield Subcode | $[n,k,d]_{q^m}$ | | $[n,\geq km-nm+n,\geq d]_q$ |
| Trace | $[n,k,d]_{q^m}$ | | $[n,\leq mk,\leq d]_q$ |
| Square | $[n,k]_q$ | | $[n,k']_q$ |
| Concatenation | $[n_1,k_1,d_1]_q$ | $[n_2,k_2,d_2]_{q^{k_1}}$ | $[n_1n_2,k_1k_2,\geq d_1d_2]_q$ |

# 7 Equivalence of Codes

In mathematics we often ask when to objects are "essentially" the same. Let us clarify what that means for codes.

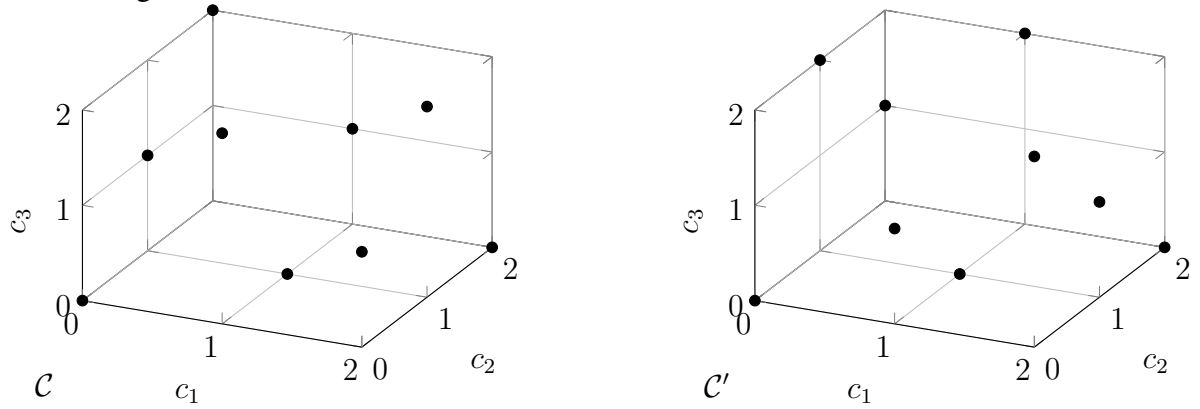
Let us consider the following two codes over $\mathbb{F}_3$: $\mathcal{C} = \langle G \rangle$ and $\mathcal{C}' = \langle G' \rangle$.

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

Then

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

While they are not the same code, their points seem to be just rotate, and all still have the same distance among each other.




In fact, we two codes are essentially the same, if we can map one code linearly to the other, in such a way that the distances between the codewords stays the same.

**Definition 7.1.** A *linear isometry* for a distance function $d$ is a linear map $\varphi : \mathbb{F}_q^n \to \mathbb{F}_q^n$, such that for all $x, y \in \mathbb{F}_q^n$ we have that

$$d(x, y) = d(\varphi(x), \varphi(y)).$$

Clearly, when dealing with a distance which is induced from a weight wt, then we can equivalently define a linear isometry $\varphi$ to be such that for all $x \in \mathbb{F}_q^n$

$$\mathrm{wt}(x) = \mathrm{wt}(\varphi(x)).$$

**Proposition 7.2.** *The linear isometries with respect to some distance function form a group with respect to the composition.*

**Exercise 7.3.** *Prove Proposition 7.2 and observe that any linear isometry is a $\mathbb{F}_q$-isomorphism.*

In our case, we are interested in the linear isometries for the Hamming metric.

**Proposition 7.4.** *The linear isometries for the Hamming metric are given by the semidirect product* $(\mathbb{F}_q^\star)^n \rtimes S_n$, *where* $S_n$ *denotes the symmetric group of degree* $n$.

*Proof.* For the first direction, we note that $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ is linear and can be written as matrix multiplication, where $\varphi(x) = xDP$, for $D = \text{diag}(d_1, \ldots, d_n)$ a diagonal matrix with entries $d_i \in \mathbb{F}_q^\star$ and $P$ an $n \times n$ permutation matrix belonging to the permutation $\sigma$. Thus,

$$\varphi(x_1, \ldots, x_n) = (d_{\sigma^{-1}(1)} x_{\sigma^{-1}(1)}, \ldots, d_{\sigma^{-1}(n)} x_{\sigma^{-1}(n)})$$

and if $x_i \neq 0$, then $x_{\sigma(i)} \neq 0$ and by multiplying with a non-zero scalar, we still have non-zero. On the other hand, if $x_i = 0$, then $x_{\sigma(i)} = 0$ and it remains zero after multiplying with some non-zero $d_j$.

For the other direction, let us assume that $\varphi$ is a linear isometry and denote by $e_i$ the standard vector having all zero entries but a 1 in position $i$. These vectors clearly span the whole $\mathbb{F}_q^n$ and hence to define a linear map $\varphi$, it is enough to know where $\varphi$ sends the basis $e_i$. In fact, any $x \in \mathbb{F}_q^n$ is such that $x = \sum_{i=1}^n e_i \lambda_i$ for $\lambda_1, \ldots, \lambda_n \in \mathbb{F}_q$. By the linearity of $\varphi$ we thus get

$$\varphi(x) = \sum_{i=1}^{n} \varphi(e_i)\lambda_i.$$

For all $e_i = (0, \ldots, 0, 1, 0 \ldots, 0)$ of Hamming weight 1, we must have

$$\varphi(e_i) \in \{\lambda(i) e_{j(i)} \mid \lambda(i) \in \mathbb{F}_q^\star, j(i) \in \{1, \ldots, n\}\}.$$

If we assign each of the $e_i$ for all $i \in \{1, \ldots, n\}$, then there exists a permutation $\sigma$, which sends $i \mapsto j(i)$, and scalar multiples $d_{j(i)} = \lambda(i)$.

In fact, if the map $i \mapsto j(i)$ is not a permutation, then there exist $i \neq i'$ with $j(i) = j(i')$ and hence $\varphi(e_i + e_{i'})$ is some multiple of $e_{j(i)}$. This, however, contradicts that $\varphi$ is an isometry: $\text{wt}_H(e_i + e_{i'}) = 2$, whereas $\text{wt}_H(\lambda e_{j(i)}) = 1$. □

Matrices of the form $DP$, for $D$ a diagonal matrix and $P$ a permutation matrix are also called monomial matrices and $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ monomial transforms.

We might sometimes be interested in the semi-linear isometries instead, as in fact we also have the automorphisms of the finite field itself.

The *semi-linear isometries* for the Hamming metric are then given by $(\mathbb{F}_q^\star)^n \rtimes (\text{Aut}(\mathbb{F}_q) \times S_n)$.

For codes, we thus define two codes to be equivalent, if there exists some (semi-) linear isometry between them.

However, if we have a linear isometry between two codes, i.e., let $\mathcal{C}, \mathcal{C}'$ be two $[n, k]_q$ linear codes and there exists a linear map $\varphi : \mathcal{C} \to \mathcal{C}'$ which preserves the weight, that is for all $c \in \mathcal{C}$ we have $\text{wt}_H(c) = \text{wt}_H(\varphi(c))$, we might apriori get other maps than the monomial transforms.

MacWilliams in her thesis [10] showed that this is not the case.

**Theorem 7.5** (Extension Theorem)**.** *Let $\mathcal{C}, \mathcal{C}'$ be two $[n,k]_q$ linear codes and there exists a linear map $\varphi : \mathcal{C} \to \mathcal{C}'$ which preserves the weight, then there exists a linear isometry $\mu : \mathbb{F}_q^n \to \mathbb{F}_q^n$ with $\mu_{|\mathcal{C}} = \varphi$.*

That is, any linear weight preserving map that we may find between two codes can be extended to a linear isometry of the whole ambient space.

**Definition 7.6.** Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes. We say that $\mathcal{C}$ is *equivalent* to $\mathcal{C}'$ if there exists a $\varphi \in (\mathbb{F}_q^\star)^n \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$ such that $\varphi(\mathcal{C}) = \mathcal{C}'$.

We say that $\mathcal{C}$ is *linearly equivalent* to $\mathcal{C}'$ if there exists a $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ such that $\varphi(\mathcal{C}) = \mathcal{C}'$.

We say that $\mathcal{C}$ is *permutation equivalent* to $\mathcal{C}'$ if there exists a $\varphi \in S_n$ such that $\varphi(\mathcal{C}) = \mathcal{C}'$.

The equivalence between the two codes also gives rise to a condition for their generator matrices.

**Proposition 7.7.** *Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes with generator matrices $G$, respectively $G'$. If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, then there exist matrices $S \in \mathbf{GL}_q(k), D = \mathrm{diag}(d_1, \ldots, d_n)$ with $d_i \in \mathbb{F}_q^\star$ and a $n \times n$ permutation matrix $P$, such that*

$$SGDP = G'.$$

This also includes the case of permutation equivalence by setting $d_i = 1$ for all $i \in \{1, \ldots, n\}$.

**Exercise 7.8.** *Prove Proposition 7.7.*

**Example 7.9.** *Let us consider $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ where $\alpha^2 = \alpha + 1$.*

*Let $\mathcal{C} = \langle G \rangle$ where*

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

*Then we may apply the permutation $\sigma = (1,2)$ to get*

$$GP = \begin{pmatrix} 0 & 1 & \alpha \\ 1 & 0 & \alpha + 1 \end{pmatrix}$$

*and $\langle GP \rangle$ is permutation equivalent to $\mathcal{C}$.*

*If we also apply the diagonal matrix $D = \mathrm{diag}(1, \alpha, \alpha + 1)$, we get*

$$GPD = \begin{pmatrix} 0 & \alpha & 1 \\ 1 & 0 & \alpha \end{pmatrix}$$

*with $\langle GPD \rangle$ is linearly equivalent to $\langle GP \rangle$ and to $\mathcal{C}$.*

*Finally, we can also apply the non-trivial automorphism of $\mathbb{F}_4$, namely $x \mapsto x^2$, to get*

$$G' = \begin{pmatrix} 0 & \alpha + 1 & 1 \\ 1 & 0 & \alpha + 1 \end{pmatrix},$$

*with $\langle G' \rangle$ is equivalent to $\langle GP \rangle, \langle GPD \rangle$ and $\mathcal{C}$.*

We also note that our definition of systematic form, i.e., there exists an information set $I$ such that we may write

$$G_I = \mathrm{Id}_k, \quad G_{I^C} = A,$$

for some $A \in \mathbb{F}_q^{k\times(n-k)}$, can now be read as: any code is permutation equivalent to a code with generator matrix

$$G' = \begin{pmatrix} \mathrm{Id}_k & A \end{pmatrix}.$$

On the other hand we also have linear isometries from $\mathcal{C}$ to itself.

**Definition 7.10** (Automorphism Group)**.** Let $\mathcal{C}$ be an $[n,k]_q$ linear code. The *automorphism group* of $\mathcal{C}$ is given by the linear isometries that map $\mathcal{C}$ to $\mathcal{C}$ :

$$\mathrm{Aut}(\mathcal{C}) = \{\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n \mid \varphi : \mathcal{C} \to \mathcal{C}\}.$$

If $\varphi$ is a linear isometry from $\mathcal{C}$ to $\mathcal{C}'$, then $\varphi^{-1}$ is a linear isometry from $\mathcal{C}'$ to $\mathcal{C}$, which gives us an easy way to construct automorphisms: let $\varphi, \psi : \mathcal{C} \to \mathcal{C}'$ be two linear isometries, then $\psi^{-1} \circ \varphi \in \mathrm{Aut}(\mathcal{C})$.

On the other hand, having $\varphi : \mathcal{C} \to \mathcal{C}'$ and $\psi : \mathcal{C} \to \mathcal{C}$ to linear isometries, then $\varphi \circ \psi$ is again a linear isometry between $\mathcal{C}$ and $\mathcal{C}'$.

**Exercise 7.11.** *Let $\varphi \in$ Aut$(\mathcal{C})$ be a permutation. Show that $\varphi \in$ Aut$(\mathcal{C}^\perp)$.*

Just like the hull, the automorphism group of a random linear code is with high probability trivial [9], i.e., $\mathrm{Aut}(\mathcal{C}) = \{\mathrm{id}\}$.

**Exercise 7.12.** *Give the automorphism group of $\mathcal{C} = \langle (1,0,0), (0,1,1)\rangle \subseteq \mathbb{F}_2^3$.*

**Exercise 7.13.** *Let $\varphi \in$ Aut$(\mathcal{C})$ be a permutation. Show that $\varphi \in$ Aut$(\mathcal{C} \cap \mathcal{C}^\perp)$.*

**Proposition 7.14.** *Let $\mathcal{C}_1, \mathcal{C}_2$ be two permutation equivalent $[n,k,d]_q$ linear codes. Then $\mathcal{C}_1^\perp$ is permutation equivalent to $\mathcal{C}_2^\perp$.*

*Proof.* Let $\sigma \in S_n$ be such that $\sigma(\mathcal{C}_1) = \mathcal{C}_2$ and denote by $P$ the permutation matrix with respect to $\sigma$. Let $G_1, G_2$ be generator matrices for $\mathcal{C}_1$, respectively $\mathcal{C}_2$, then there exist a $S \in \mathrm{GL}_q(k)$ such that $SG_1P = G_2$.

Let $H_1, H_2$ be the parity-check matrices for $\mathcal{C}_1$, respectively $\mathcal{C}_2$. Since $G_2H_2^\top = 0$, we also have $G_1PH_2^\top = G_1(H_2P^\top)^\top = 0$. This implies that $H_2P^\top$ is a parity-check matrix for $\mathcal{C}_1$ and hence $H_2 = S'H_1P$, for some $S \in \mathrm{GL}_q(n-k)$. Thus, $\sigma(\mathcal{C}_1^\perp) = \mathcal{C}_2^\perp$.

□

**Exercise 7.15.** *Let $\mathcal{C}, \mathcal{C}'$ be linearly equivalent codes. Show that $\mathcal{C}^\perp$ is linearly equivalent to $\mathcal{C}'^\perp$.* Hint: *Use the fact that $GH^\top = 0$ and $SGPD = G'$.*

For two permutation equivalent codes, their hulls are also permutation equivalent.

**Proposition 7.16.** *Let $\mathcal{C}_1, \mathcal{C}_2$ be two permutation equivalent $[n,k,d]_q$ linear codes. Then $\mathcal{H}(\mathcal{C}_1)$ is permutation equivalent to $\mathcal{H}(\mathcal{C}_2)$.*

*Proof.* Let $\sigma \in S_n$ be such that $\sigma(\mathcal{C}_1) = \mathcal{C}_2$ and denote by $P$ the permutation matrix with respect to $\sigma$. Let $G_1$ be a generator matrix for $\mathcal{C}_1$ and $H_1$ be a parity-check matrix for $\mathcal{C}_1$.

Recall that $G_1P$ is a generator matrix for $\mathcal{C}_2$ and $H_1P$ is a parity-check matrix for $\mathcal{C}_2$. Finally, we have that

$$\mathcal{H}(\mathcal{C}_2) = \ker\left(\begin{pmatrix} G_2 \\ H_2 \end{pmatrix}^\top\right) = \ker\left(\begin{pmatrix} G_1P \\ H_1P \end{pmatrix}^\top\right) = \ker\left(\begin{pmatrix} G_1 \\ H_1 \end{pmatrix}^\top\right) P.$$

□

To determine whether two codes are equivalent is important, especially when claiming one has found a new construction of a code. In this case, one should first check whether this new family of codes is not equivalent to an already known family.

However, determining whether two codes are equivalent or not is not an easy task. We might instead look for *invariants*, i.e., properties of a code $\mathcal{C}$ that remain the same for $\varphi(\mathcal{C})$.

## 7.1 Invariants

There are several parameters or properties of equivalent codes which remain invariant. Clearly, equivalent codes have the same length, dimension and minimum distance.

But we can also find more such invariants.

**Definition 7.17** (Weight Enumerator)**.** Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code. For any $w \in \{1, \dots, n\}$, let us denote by $A_w(\mathcal{C}) = |\{c \in \mathcal{C} \mid \mathrm{wt}_H(c) = w\}|$ the *weight enumerator* of $\mathcal{C}$.

**Proposition 7.18.** *Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ be linearly equivalent codes, then for all $w \in \{1, \dots, n\}$ we have that*

$$A_w(\mathcal{C}_1) = A_w(\mathcal{C}_2).$$

*Proof.* Since $\mathcal{C}_1$ is linearly equivalent to $\mathcal{C}_2$, there exists some isometry $\varphi : \mathcal{C}_1 \to \mathcal{C}_2$. Thus, if we consider the set

$$S_w(\mathcal{C}_1) = \{c \in \mathcal{C}_1 \mid \mathrm{wt}_H(c) = w\}$$

then

$$\begin{aligned} \varphi(S_w(\mathcal{C}_1)) &= \{\varphi(c) \mid c \in \mathcal{C}_1, \mathrm{wt}_H(c) = w\} \\ &= \{c' \in \mathcal{C}_2 \mid \mathrm{wt}_H(c') = w\} = S_w(\mathcal{C}') \end{aligned}$$

and hence they have the same size. □

Note that the other direction is not true: We can have codes with the same weight enumerator, which are not linearly equivalent!

**Example 7.19.** *Let us consider $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$ with $\alpha^2 = \alpha + 1$. The two codes $\mathcal{C}_1 = \langle G_1 \rangle, \mathcal{C}_2 = \langle G_2 \rangle$ with*

$$G_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \alpha \end{pmatrix}, G_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & \alpha+1 \end{pmatrix}$$

*have the same weight enumerators. In fact, all codewords of $\mathcal{C}_1$, respectively $\mathcal{C}_2$, either have no zero, 4 zeros in $\{1,2,3,4\}$, 3 zeros in $\{5,6,7\}$, 2 zeros in $\{8,9\}$ or 1 zero in $\{10\}$, but no mixed zeros between these index sets. Thus,*

$$\begin{aligned} A_0(\mathcal{C}_1) = A_0(\mathcal{C}_2) &= 1, \\ A_1(\mathcal{C}_i) = A_2(\mathcal{C}_i) = A_3(\mathcal{C}_i) = A_4(\mathcal{C}_i) = A_5(\mathcal{C}_i) &= 0, \\ A_6(\mathcal{C}_1) = A_6(\mathcal{C}_2) &= 3, \\ A_7(\mathcal{C}_1) = A_7(\mathcal{C}_2) &= 3, \\ A_8(\mathcal{C}_1) = A_8(\mathcal{C}_2) &= 3, \\ A_9(\mathcal{C}_1) = A_9(\mathcal{C}_2) &= 3, \\ A_{10}(\mathcal{C}_1) = A_{10}(\mathcal{C}_2) &= 3. \end{aligned}$$

*However, there is no linear equivalence between $\mathcal{C}_1$ and $\mathcal{C}_2$. To see this, let us assume that there exists a $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$, which is such that $\varphi(\mathcal{C}_1) = \mathcal{C}_2$. Such $\varphi$ would need to send the weight 7 codewords of $\mathcal{C}_1$ to the weight 7 codewords of $\mathcal{C}_2$, that is if $(1,1,1,1,0,0,0,1,1,\alpha) = x$ and $(1,1,1,1,0,0,0,1,1,\alpha+1) = y$, then $\varphi(x) \in \{y, \alpha y, (\alpha+1)y\}$.*

*If $\varphi(x) = y$, then $\varphi$ would be a permutation, except for the index which gets sent to $y_{10} = \alpha + 1$, and the index which sends $x_{10}$ somewhere, i.e., if $\varphi = DP$, for $D = diag(d_1, \ldots, d_n)$, $P$ a permutation matrix belonging to the permutation $\sigma \in S_n$, then $d_{10} \neq 1$ and $d_{\sigma^{-1}(10)} \neq 1$.*

*The same $\varphi$ also needs to send the codewords of weight 6 to each other. Here the two sets are the same for $\mathcal{C}_1, \mathcal{C}_2$, implying that $\varphi$ can only have $d_5 = \cdots = d_{10}$, a contradiction.*

*The cases $\varphi(x) \in \{\alpha y, (\alpha+1)y\}$ work similarly.*

Another invariant are the generalized weights. For this, we need to introduce the support of a code.

**Definition 7.20** (Support of a Code)**.** Let $\mathcal{C}$ be a $[n,k,d]_q$ linear code. The *support* of $\mathcal{C}$ is defined as

$$\text{Supp}_H(\mathcal{C}) = \{i \in \{1,\ldots,n\} \mid \exists c \in \mathcal{C} : c_i \neq 0\}.$$

Clearly, for a non-degenerate code, the support will be full, i.e., $\{1,\ldots,n\}$, however, as soon as we go to subcodes of $\mathcal{C}$, this will change. Similar to how the weight of a vector is the size of its support, we may define the *weight* of a code as the size of its support.

**Definition 7.21.** Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. The *weight* of $\mathcal{C}$ is given by

$$\text{wt}_H(\mathcal{C}) = |\text{Supp}_H(\mathcal{C})|.$$

Again, if $\mathcal{C}$ is non-degenerate then $\mathrm{wt}_H(\mathcal{C}) = n$.

Clearly, the weight of a code is also an invariant for code equivalence: if $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathrm{wt}_H(\mathcal{C}) = \mathrm{wt}_H(\mathcal{C}')$.

We may now consider the smallest weights of any subcode:

**Definition 7.22.** Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code and let $r \in \{1, \ldots, k\}$. The $r$th *generalized weight* of $\mathcal{C}$ is given by

$$d_r(\mathcal{C}) = \min\{\mathrm{wt}_H(\mathcal{D}) \mid \mathcal{D} \subset \mathcal{C}, \dim(\mathcal{D}) = r\}.$$

If $r = 1$, we are asking for the smallest weight of any $c \in \mathcal{C}$, i.e., the first generalized weight $d_1(\mathcal{C})$ is the minimum distance $d_H(\mathcal{C})$.

On the other hand, if $r = k$, we are asking for the weight of the whole code, i.e., $d_k(\mathcal{C}) = \mathrm{wt}_H(\mathcal{C})$.

**Example 7.23.** *Let $\mathcal{C} = \langle G \rangle \subset \mathbb{F}_2^4$, where*

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

*Then $d_1 = d_H(\mathcal{C}) = 1$ as $\mathcal{D}_1 = \langle (0, 1, 0, 0) \rangle$ has the smallest weight $wt_H(\mathcal{D}_1) = 1$. $d_2 = 3$ as $\mathcal{D}_2 = \langle (0, 1, 0, 0), (0, 0, 1, 1) \rangle$ has the smallest weight $wt_H(\mathcal{D}_2) = 3$ and finally $d_3 = wt_H(\mathcal{C}) = 4$.*

**Exercise 7.24.** *Show that generalized weights are strictly increasing, that is for $r \in \{1, \ldots, k-1\}$ we have $d_r(\mathcal{C}) < d_{r+1}(\mathcal{C})$.*

**Proposition 7.25.** *Let $\mathcal{C}_1, \mathcal{C}_2$ be $[n, k, d]_q$ linear codes, which are linearly equivalent.*
*For all $r \in \{1, \ldots, k\}$ we have that*

$$d_r(\mathcal{C}_1) = d_r(\mathcal{C}_2).$$

*Proof.* Let $\varphi \in (\mathbb{F}_q^\times)^n \rtimes S_n$ be such that $\varphi(\mathcal{C}_1) = \mathcal{C}_2$ and let $\mathcal{D}$ be any subcode of $\mathcal{C}_1$, then $\varphi(\mathcal{D})$ is a subcode of $\mathcal{C}_2$.

As $\mathrm{wt}_H(\mathcal{D}) = \mathrm{wt}_H(\varphi(\mathcal{D}))$, we immediately get

$$\begin{aligned} d_r(\mathcal{C}_1) &= \min\{\mathrm{wt}_H(\mathcal{D}) \mid \mathcal{D} \subset \mathcal{C}_1, \dim(\mathcal{D}) = r\} \\ &= \min\{\mathrm{wt}_H(\varphi(\mathcal{D})) \mid \varphi(\mathcal{D}) \subset \varphi(\mathcal{C}_1), \dim(\varphi(\mathcal{D})) = r\} \\ &= d_r(\varphi(\mathcal{C}_1)) = d_r(\mathcal{C}_2). \end{aligned}$$

□

A last invariant is the size of the automorphism group.

**Proposition 7.26.** *Let $\mathcal{C}_1, \mathcal{C}_2$ be two linearly equivalent $[n,k,d]_q$ linear codes. Then*

$$|\mathit{Aut}(\mathcal{C}_1)| = |\mathit{Aut}(\mathcal{C}_2)|.$$

*Proof.* If $\varphi \in (\mathbb{F}_q^\times)^n \rtimes S_n$ is such that $\varphi(\mathcal{C}_1) = \mathcal{C}_2$, then for any $\psi \in \mathrm{Aut}(\mathcal{C}_1)$ we have that

$$\psi' = \varphi \circ \psi \circ \varphi^{-1} \in \mathrm{Aut}(\mathcal{C}_2).$$

□

## 7.2 Closure

Given two linearly equivalent codes, we can construct two new codes, which are now permutation equivalent.

For this we introduce the *closure* of a code.

**Definition 7.27.** Let $\mathcal{C}$ be an $[n,k]_q$ linear code, let $\alpha \in \mathbb{F}_q$ be a primitive element and denote by $\lambda = (1, \alpha, \ldots, \alpha^{q-2}) \in \mathbb{F}_q^{q-1}$. The *closure* of $\mathcal{C}$ is given by the Kronecker product $\lambda \otimes \mathcal{C}$.

The new code is now of length $n(q-1)$ and still of dimension $k$. In fact, if $G$ is a generator matrix of $\mathcal{C}$, then $\lambda \otimes G$ is a generator matrix of $\lambda \otimes \mathcal{C}$.

**Proposition 7.28.** *Let $\mathcal{C}_1, \mathcal{C}_2$ be two linearly equivalent $[n,k]_q$ linear codes. Then $\lambda \otimes \mathcal{C}_1$ is permutation equivalent to $\lambda \otimes \mathcal{C}_2$.*

*Proof.* Let $\varphi = DP$, with $D = \mathrm{diag}(d_1, \ldots, d_n)$ and $P$ a $n \times n$ permutation matrix, such that $\varphi(\mathcal{C}_1) = \mathcal{C}_2$.

If $G_1 = \begin{pmatrix} g_1^\top & \cdots & g_n^\top \end{pmatrix}$ is a generator matrix for $\mathcal{C}_1$ then

$$\lambda \otimes G_1 = \begin{pmatrix} g_1^\top & \alpha g_1^\top & \cdots & \alpha^{q-2} g_1^\top & \cdots & g_n^\top & \alpha g_n^\top & \cdots & \alpha^{q-2} g_n^\top \end{pmatrix}$$

is a generator matrix of $\lambda \otimes \mathcal{C}_1$.

We note that multiplying with $d_i$ is a permutation in $\mathbb{F}_q^\star$, that is $\sigma_i : \mathbb{F}_q^\star \to \mathbb{F}_q^\star, x \mapsto x d_i$ can be seen as $\sigma_i \in S_{q-1}$. Thus, multiplying column $g_i$ with $d_i$, means $(d_i g_i^\top, d_i \alpha g_i^\top, \ldots, d_i \alpha^{q-2} g_i^\top) = \sigma(g_i^\top, \alpha g_i^\top, \ldots, \alpha^{q-2} g_i^\top)$. Hence the scalars $d_i$ are introducing permutations $\sigma_i$ within the $n$ blocks of length $m$.

The permutation $P$ instead shifts around these blocks, that is if $\sigma$ is the permutation corresponding to $P$ and $\sigma$ sends the index $i$ to $j$, then we have to send the $i$th block to the $j$th block.

Thus,

$$Q = \begin{pmatrix} P_1 & & \\ & \ddots & \\ & & P_n \end{pmatrix} (\mathrm{Id}_{q-1} \otimes P) \in S_{n(q-1)}$$

is such that $(\lambda \otimes G_1)Q$ is a generator matrix of $\lambda \otimes \mathcal{C}_2$. □

The problem of finding a permutation between two random codes can be solved in quasi-polynomial time, using a reduction to the Graph Isomorphism (GI) problem and Babai's quasi-polynomial time solver for GI [1].

This works only with high probability, as it requires the code to have a trivial hull.

When first reducing linearly equivalent codes to two permutation equivalent codes, we cannot reduce them further to GI, as the closure of codes is in fact self-orthogonal for $q \geq 4$.

**Proposition 7.29.** *If $q \geq 4$, then $\lambda \otimes \mathcal{C}$ is self-orthogonal.*

*Proof.* Recall that a code $\mathcal{D}$ is self-orthogonal if $\mathcal{D} \subseteq \mathcal{D}^\perp$, thus $\mathcal{H}(\mathcal{D}) = \mathcal{D}$. We have also seen that if $D\ G$ is a generator matrix of $\mathcal{D}$, then $GG^\top = 0$ implies that $\mathcal{D}$ is self-orthogonal.

To understand the hull of the closure, we thus have to compute

$$(\lambda \otimes G)(\lambda \otimes G)^\top = \begin{pmatrix} g_1^\top & \alpha g_1^\top & \cdots & \alpha^{q-2} g_n^\top \end{pmatrix} \begin{pmatrix} g_1 \\ \alpha g_1 \\ \vdots \\ \alpha^{q-2} g_n \end{pmatrix}.$$

One can easily check that

$$(\lambda \otimes G)(\lambda \otimes G)^\top = \lambda\lambda^\top GG^\top.$$

While we assumed that for random $G$ we have that $GG^\top$ is full rank, we also have that $\lambda\lambda^\top = 0$, as

$$\lambda\lambda^\top = \sum_{i=0}^{q-2} \alpha^{2i} = \sum_{\beta \in \mathbb{F}_q^\star} \beta^2 = 0,$$

by Lemma 3.23, unless $q = 2, 3$.

□

On the other hand, if $q < 4$, then $\lambda \otimes \mathcal{C}$ has with high probability a trivial hull.

**Example 7.30.** *Let us consider $\mathbb{F}_4 = \mathbb{F}_2(\alpha)$, where $\alpha^2 = \alpha + 1$. Let $\mathcal{C} = \langle G \rangle \subset \mathbb{F}_4^3$, where*

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

*Let $D = diag(\alpha, 1, \alpha)$ and $P$ be the permutation matrix corresponding to $\sigma = (1, 3)$, then*

$$GDP = \begin{pmatrix} \alpha + 1 & 0 & \alpha \\ \alpha & 1 & 0 \end{pmatrix}.$$

*Bringing this into systematic form, we get a linearly equivalent code $\mathcal{C}'$ generated by*

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

*Let* $\lambda = (1, \alpha, \alpha+1)$. *The closure of* $\mathcal{C}$ *is generated by*

$$\lambda \otimes G = \begin{pmatrix} 1 & \alpha & \alpha+1 & 0 & 0 & 0 & \alpha & \alpha+1 & 1 \\ 0 & 0 & 0 & 1 & \alpha & \alpha+1 & 1 & \alpha & \alpha+1 \end{pmatrix},$$

*while the closure of* $\mathcal{C}'$ *is generated by*

$$\lambda \otimes G' = \begin{pmatrix} 1 & \alpha & \alpha+1 & 0 & 0 & 0 & \alpha+1 & 1 & \alpha \\ 0 & 0 & 0 & 1 & \alpha & \alpha+1 & 1 & \alpha & \alpha+1 \end{pmatrix}.$$

*We can find*

$$Q = \begin{pmatrix} 0 & 0 & 1 & & & & & & \\ 1 & 0 & 0 & & 0 & & & 0 & \\ 0 & 1 & 0 & & & & & & \\ & & & 1 & 0 & 0 & & & \\ & 0 & & 0 & 1 & 0 & & 0 & \\ & & & 0 & 0 & 1 & & & \\ & & & & & & 0 & 0 & 1 \\ & 0 & & & 0 & & 1 & 0 & 0 \\ & & & & & & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & Id_3 \\ 0 & Id_3 & 0 \\ Id_3 & 0 & 0 \end{pmatrix} \in S_9$$

*which is such that* $S(\lambda \otimes G)Q = \lambda \otimes G'$, *for some* $S \in GL_q(k)$.

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes.

- The *linear isometries* in the Hamming metric are $(\mathbb{F}_q^\star)^n \rtimes S_n$.
- $\mathcal{C}, \mathcal{C}'$ are *linearly equivalent* if there exists a $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ such that $\varphi(\mathcal{C}) = \mathcal{C}'$.
- $\mathcal{C}, \mathcal{C}'$ are *permutation equivalent* if there exists a $\varphi \in S_n$ such that $\varphi(\mathcal{C}) = \mathcal{C}'$.
- If $\mathcal{C}$ and $\mathcal{C}'$ are linearly equivalent, then their duals are linearly equivalent.
- The *automorphism group* of $\mathcal{C}$ are the linear isometries $\varphi : \mathcal{C} \to \mathcal{C}$.
- The *weight enumerator* $A_w(\mathcal{C})$ is the amount of codewords in $\mathcal{C}$ of weight $w$.
- The weight enumerator is invariant for equivalent codes.
- The $r$th *generalized weight* is $d_r(\mathcal{C}) = \min\{\mathrm{wt}_H(\mathcal{D}) \mid \mathcal{D} \subseteq \mathcal{C}, \dim(\mathcal{D}) = r\}$.
- The $r$th generalized weight is invariant for equivalent codes.
- Let $\alpha$ be a primitive element in $\mathbb{F}_q$, the *closure* of $\mathcal{C}$ is $(1, \alpha, \ldots, \alpha^{q-2}) \otimes \mathcal{C}$.
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, then their closures are permutation equivalent.

# 8 Cyclic Codes

Another family of interesting codes is that of cyclic codes. From a practical point of view, cyclic codes admit a very compact representation and enjoy very efficient decoders.

From the theoretical side, they are full of algebraic structure as we describe next.
First, let us introduce the shift of vectors to the right:

**Definition 8.1.** Let us define the following map

$$\begin{aligned}\sigma : \mathbb{F}_q^n &\to \mathbb{F}_q^n,\\ (x_0, x_1, \ldots, x_{n-1}) &\mapsto (x_{n-1}, x_0, \ldots, x_{n-2}).\end{aligned}$$

Then $\sigma$ is called a *cyclic shift.*

Attention, in this chapter, we will mostly start indexing our vectors at 0 instead of 1.
We may also write $\sigma^i$ for some $i \geq 1$ to denote

$$\underbrace{\sigma \circ \cdots \circ \sigma}_{i \text{ times}}.$$

**Definition 8.2.** Let $\mathcal{C}$ be an $[n, k]_q$ linear code. Then $\mathcal{C}$ is called *cyclic* if $\sigma(\mathcal{C}) = \mathcal{C}$.

Thus, if for all $c \in \mathcal{C}$ we have that $\sigma(c) = c' \in \mathcal{C}$, then also $\sigma^2(c) = \sigma(c') \in \mathcal{C}$. We conclude that for any $i$, we have that $\sigma^i(c) \in \mathcal{C}$. Clearly, $\sigma^n(c) = \sigma^0(c) = c$, hence the power of $\sigma$ should always be treated modulo $n$.

Clearly all (non-degenerate) trivial codes are cyclic, or equivalent to a cyclic code.

In fact, for $k = 0$ we have that $\{0\}$ is cyclic, similarly for $k = n$ we have that $\mathbb{F}_q^n$ is cyclic and if $k = 1$, then the code generated by $(c, \ldots, c)$ is cyclic, similarly for $k = n - 1$, we have that the code generated by $G = \begin{pmatrix} & 1 \\ \mathrm{Id}_{n-1} & \vdots \\ & 1 \end{pmatrix}$ is cyclic.

**Example 8.3.** *Let us consider the code generated by $G = (1, 2)$ in $\mathbb{F}_5$, then clearly $(2, 1) \notin \langle G \rangle$, and $\langle G \rangle$ is not cyclic, however it is equivalent to the code generated by $G = (1, 1)$.*

We have also seen some non-trivial cyclic codes:

**Lemma 8.4.** *Let $a \in \mathbb{F}_q$ be a primitive element and define $\alpha = (1, a, \ldots, a^{q-2})$. Then the primitive Reed-Solomon code $\mathcal{RS}_{q,n,k}(\alpha)$ is cyclic for all $k \leq q - 1$.*

*Proof.* Let $f(x) \in \mathbb{F}_q[x]$ be an arbitrary polynomial of degree $\deg(f) < k$ and consider the codeword

$$c = (f(1), f(a), \ldots, f(a^{q-2})) \in \mathcal{RS}_{q,n,k}(\alpha).$$

Let us define $g(x) = f(a^{q-2}x)$, then we clearly have that

$$g(a^i) = f(a^{i-1})$$

for all $i \in \{0, \ldots, q-2\}$ and hence

$$c' = (g(1), g(a), \ldots, g(a^{q-2})) = (f(a^{q-2}), f(1), \ldots, f(a^{q-3})) = \sigma(c).$$

Since $\deg(g) < k$, we have that $c' = \sigma(c) \in \mathcal{RS}_{q,n,k}(\alpha)$. □

**Lemma 8.5.** *Let $\alpha = (0, 1, \ldots, p-1)$ then for any $k \leq p-1$, we have that the primitive Reed-Solomon code $\mathcal{RS}_{p,n,k}(\alpha)$ is cyclic.*

**Exercise 8.6.** *Prove Lemma 8.5 using $g(x) = f(x-1)$.*

**Exercise 8.7.** *Is every Reed-Solomon code a cyclic code?*

## 8.1 Polynomial Representation

To understand the algebraic structure, we associate to any vector $(c_0, \ldots, c_{n-1})$ the polynomial $c(x) = \sum_{i=}^{n-1} c_i x^i$. We have to restrict the degree of $c(x) \in \mathbb{F}_q[x]$ to be $n-1$ and note that $\mathbb{F}_q[x]/(x^n - 1)$ is isomorphic to

$$\{f(x) \in \mathbb{F}_q[x] \mid \deg(f) < n\}.$$

**Proposition 8.8.** *Let $n$ be a positive integer, then*

$$\begin{aligned} \varphi : \mathbb{F}_q[x]/(x^n - 1) &\to \mathbb{F}_q^n, \\ c(x) = \sum_{i=0}^{n-1} c_i x^i &\mapsto (c_0, \ldots, c_{n-1}) \end{aligned}$$

*is a $\mathbb{F}_q$-vector space isomorphism.*

**Exercise 8.9.** *Prove Proposition 8.8.*

Although they are isomorphic as vector spaces, $\mathbb{F}_q[x]/(x^n - 1)$ enjoys some more algebraic structure: it is a ring!

**Lemma 8.10.** *The cyclic shift $\sigma : \mathbb{F}_q^n \to \mathbb{F}_q^n$ corresponds in $\mathbb{F}_q[x]/(x^n - 1)$ to multiplying with $x$.*

*Proof.* Let $\varphi(c(x)) = (c_0, \ldots, c_{n-1}) = c$, then $\sigma(c) = (c_{n-1}, c_0, \ldots, c_{n-2})$ has the preimage $\sigma(c)(x) = \sum_{i=0}^{n-1} c_{i-1} x^i$, where the indices are modulo $n$. On the other hand

$$xc(x) = \sum_{i=0}^{n-1} c_i x^{i+1} = \sum_{i=0}^{n-1} c_{i-1} x^i = \sigma(c)(x).$$

□

With this we have a correspondence between ideals in $\mathbb{F}_q[x]/(x^n-1)$ and cyclic codes in $\mathbb{F}_q^n$.

**Proposition 8.11.** *$\varphi$ introduces a 1-to-1 correspondence between ideals over $\mathbb{F}_q[x]/(x^n-1)$ and cyclic codes in $\mathbb{F}_q^n$.*

*Proof.* Let us consider an ideal $I$ in $\mathbb{F}_q[x]/(x^n-1)$ generated by $g(x) = \sum_{i=0}^{n-1} g_i x^i$ and denote $\varphi(g(x)) = g \in \mathbb{F}_q^n$.

Thus any element of $I$ can be written as

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

If $\lambda(x) = \sum_{i=0}^{n-1} \lambda_i x^i$, we can write

$$a(x) = \lambda_0 g(x) + \lambda_1(xg(x)) + \cdots + \lambda_{n-1}(x^{n-1}g(x))$$

and thus

$$\varphi(a(x)) = \lambda_0 g + \lambda_1 \sigma(g) + \cdots + \sigma^{n-1}(g).$$

Since $g(x) \in I$, we have that $\varphi(g(x)) = g \in \mathcal{C}$ and since $\mathcal{C}$ is cyclic, we have that $\sigma^i(g) \in \mathcal{C}$, thus also $\varphi(a(x)) \in \mathcal{C}$ for any $a(x) \in I$.

The other direction works similarly. □

Even more is true, we can only focus on the factors of $(x^n-1)$ in $\mathbb{F}_q[x]$. We state this lemma without proof, but this result follows straight from some facts from algebra:

- $\mathbb{F}_q[x]$ is a principal ideal ring (meaning each ideal $I \subseteq \mathbb{F}_q[x]$ can be written as $I = \langle a(x) \rangle$),
- for any ideal $I = \langle a(x) \rangle \subseteq \mathbb{F}_q[x]$, $\mathbb{F}_q[x]/I$ is also a principle ideal ring and any ideal $J = \langle \overline{b(x)} \rangle$ with $\overline{b(x)} \in \mathbb{F}_q[x]/I$ is such that $b(x) \in \mathbb{F}_q[x]$ with $b(x) \mid a(x)$.

**Lemma 8.12.** *There is a 1-to-1 correspondence between ideals in $\mathbb{F}_q[x]/(x^n-1)$ and monic (with leading coefficient equal to 1) divisors of $(x^n-1)$.*

**Example 8.13.** *Let us consider $\mathbb{F}_2$ and $n = 7$. We note that*

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

*The ideal generated by the polynomial $g(x) = x^3 + x + 1$ is given by*

$$\begin{aligned} I = \{0, & x^3+x+1, x^4+x^2+x, x^4+x^3+x^2+1, x^5+x^3+x^2, x^5+x^2+x+1, \\ & x^5+x^4+x^3+x, x^5+x^4+1, x^6+x^4+x^3, x^6+x^4+x+1, x^6+x^3+x^2+x, \\ & x^6+x^5+x^4+x^2, x^6+x^2+1, x^6+x^5+x^4+x^3+x+1, x^6+x^5+x, x^6+x^5+x^3+1\}. \end{aligned}$$

*Hence*

$$\begin{aligned}\varphi(I) = \{&(0,0,0,0,0,0,0),(1,1,0,1,0,0,0),(0,1,1,0,1,0,0),(1,0,1,1,1,0,0),(0,0,1,1,0,1,0),\\&(1,1,1,0,0,1,0),(0,1,0,1,1,1,0),(1,0,0,0,1,1,0),(0,0,0,1,1,0,1),(1,1,0,0,1,0,1),\\&(0,1,1,1,0,0,1),(0,0,1,0,1,1,1),(1,0,1,0,0,0,1),(1,1,0,1,1,1,1),(0,1,0,0,0,1,1),\\&(1,0,0,1,0,1,1)\}.\end{aligned}$$

*We can generate this code using*

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

Since we can see cyclic codes as ideals in $\mathbb{F}_q[x]/(x^n-1)$, we can also consider the *generator polynomial* $g(x)$ of a cyclic code. As we have seen above, it is enough to consider $g(x)$ which are such that $g(x) \mid (x^n-1)$.

**Definition 8.14** (Generator Polynomial)**.** The *generator polynomial* of a cyclic code $\mathcal{C} \subset \mathbb{F}_q^n$ is the monic generator of minimal degree of the corresponding ideal in $\mathbb{F}_q[x]/(x^n-1)$.

This generator polynomial is in fact unique:

**Exercise 8.15.** *Let $g(x)$ be a polynomial in $\mathbb{F}_q[x]$ with $g(x) \mid (x^n-1)$. If $\langle g(x)\rangle = \langle g'(x)\rangle$ for some $g'(x) \in \mathbb{F}_q[x]$, show that $g'(x) = \lambda g(x)$ for some $\lambda \in \mathbb{F}_q^\star$.*

In our previous example, we considered $g(x) = x^3+x+1$, a factor of $(x^7-1)$ and constructed a code of dimension $7-3=4$. Additionally, the generator matrix in our example has a lot of structure, it is built as

$$G = \begin{pmatrix} \varphi(g(x)) \\ \varphi(xg(x)) \\ \vdots \\ \varphi(x^{n-\deg(g)}g(x)) \end{pmatrix} = \begin{pmatrix} g \\ \sigma(g) \\ \vdots \\ \sigma^{n-\deg(g)}(g) \end{pmatrix}.$$

This is true in general:

**Theorem 8.16.** *Let $\mathcal{C}$ be an $[n,k]_q$ linear cylcic code. Then there exists a* generator $c \in \mathcal{C}$*, such that $c, \sigma(c), \ldots, \sigma^{k-1}(c)$ generate $\mathcal{C}$.*

*Proof.* Let us consider a generator matrix $G$ of $\mathcal{C}$ in systematic form. Without loss of generality, we may assume $G = \begin{pmatrix} A & \mathrm{Id}_k \end{pmatrix}$. Thus, there exists a codeword $c \in \mathcal{C}$ (e.g. the first row of $G$) which has zeros in the last $k-1$ positions.

$$c = (c_0, \ldots, c_{n-k}, 0 \ldots, 0).$$

Note that $c, \sigma(c), \ldots, \sigma^{k-1}(c)$ are all linearly independent, as stacked together we get the matrix

$$G' = \begin{pmatrix} c_0 & \cdots & c_{n-k} & 0 & \cdots & 0 \\ 0 & c_0 & \cdots & c_{n-k} & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & c_0 & \cdots & c_{n-k} \end{pmatrix}$$

which has full rank $k$.

Thus $\langle G' \rangle = \mathcal{C}$ and $c$ is a generator of $\mathcal{C}$. □

The described generator $c$ is such that $c(x) = \sum_{i=0}^{n-1} c_i x^i$ has degree $\deg(c) \leq n - k$ and this polynomial turns out to be a generator for $\varphi^{-1}(\mathcal{C})$.

**Exercise 8.17.** *Let $\mathcal{C}$ be an $[n,k]_q$ linear cylcic code. Let $c \in \mathcal{C}$ be a generator. Show that $\langle \varphi^{-1}(c) \rangle = \varphi^{-1}(\mathcal{C})$ in $\mathbb{F}_q[x]/(x^n - 1)$,*

Hence, given a factor $g(x)$ of $(x^n - 1)$, we know how to construct a generator matrix $G$ for the code $\varphi(\langle g(x) \rangle)$, which is *circulant*.

**Definition 8.18.** Let $A \in \mathbb{F}_q^{k \times n}$ have rows $a_i$ for $i \in \{1, \ldots, k\}$. Then $A$ is called *circulant*, if $a_i = \sigma(a_{i-1})$, or equivalently $a_i = \sigma^{i-1}(a_1)$.

Hence the matrix $A$ is completely determined by a single row, e.g. $a_1$. This has important benefits, e.g. in storage.

**Corollary 8.19.** *Let $g(x) \in \mathbb{F}_q[x]$ be a factor of $x^n - 1$. Then the corresponding cyclic code $\mathcal{C} = \varphi(\langle g(x) \rangle$ is a $[n, n - \deg(g)]_q$ linear code.*

**Exercise 8.20.** *How many cyclic codes over $\mathbb{F}_3$ of length $4$ exist?*

## 8.2 Duality

How will duals of cyclic codes behave? We start by observing the behavior of the cyclic shift in an inner product.

**Lemma 8.21.** *Let $x, y \in \mathbb{F}_q^n$, then $\langle x, \sigma(y) \rangle = \langle \sigma^{-1}(x), y \rangle$.*

*Proof.* We write this out as

$$\langle x, \sigma(y) \rangle = \sum_{i=0}^{n-1} x_i y_{i+1} = \sum_{i=0}^{n-1} x_{i-1} y_i = \langle \sigma^{-1}(x), y \rangle.$$

□

**Example 8.22.** *Let us consider again $\mathbb{F}_2$ and $n = 7$ and the code generated by*

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

*We may bring this in systematic form to get*

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

*Its dual code is then generated by*

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

*and we can see that if $c' = (1, 1, 1, 0, 0, 1, 0)$ the rows of $H$ consist of $\sigma^i(c')$. Thus also $\mathcal{C}^\perp$ is cyclic.*

**Theorem 8.23.** *The dual of a cyclic code is cyclic.*

*Proof.* Let $\mathcal{C}$ be a cyclic code and let $c \in \mathcal{C}, c' \in \mathcal{C}^\perp$ be arbitrary. Then due to Lemma 8.21, we get

$$\langle c, \sigma(c') \rangle = \langle \sigma^{-1}(c), c' \rangle$$

and since $\sigma^{-1}(c) \in \mathcal{C}$, we get for any $c \in \mathcal{C}, c' \in \mathcal{C}^\perp$ that $\langle c, \sigma(c') \rangle = 0$, that is $\sigma(c') \in \mathcal{C}^\perp$. □

We can also describe $\mathcal{C}^\perp$ knowing the generator polynomial of $\mathcal{C}$.

**Definition 8.24.** Let $f(x) \in \mathbb{F}_q[x]$ be a polynomial of degree $d$, i.e., $f(x) = \sum_{i=0}^{d} f_i x^i$. The *reciprocal* $\tilde{f}(x)$ of $f(x)$ is obtained by reversing the order of the coefficients of $f(x)$, that is

$$\tilde{f}(x) = x^d f(1/x)$$

or equivalently $\varphi(f(x)) = (f_0, \ldots, f_d)$ then $\varphi(\tilde{f}(x)) = (f_d, \ldots, f_0)$.

**Example 8.25.** *Let us consider again $q = 2, n = 7$ and $h(x) = x^4 + x^2 + x + 1$, then the reciprocal of $h(x)$ is given by*

$$\tilde{h}(x) = x^4 + x^3 + x^2 + 1.$$

**Theorem 8.26.** *Let $g(x), h(x)$ be two polynomials in $\mathbb{F}_q[x]$ of degree $n - k$, respectively $k$, with $g(x)h(x) = x^n - 1$. Then their respective codes are dual. That is $\varphi(\langle g(x) \rangle)^\perp = \varphi(\langle \tilde{h}(x) \rangle)$.*

*Proof.* Let us denote $\mathcal{C} = \varphi(\langle g(x)\rangle)$ and $\mathcal{C}' = \langle \tilde{h}(x)\rangle)$. We first note that since $g(x)h(x) = x^n - 1$, we get

$$\deg(g) + \deg(h) = \deg(g) + \deg(\tilde{h}) = n.$$

Thus,

$$\dim(\mathcal{C}) + \dim(\mathcal{C}') = n$$

and we are left with showing their orthogonality.

Let $\deg(h) = k$. If $k = 0$, then $h(x) = \tilde{h}(x) = 1$ and $g(x) = x^n - 1$ and their codes $\mathcal{C}' = \mathbb{F}_q^n, \mathcal{C} = \{0\}$ are clearly dual.

Let us assume $k \geq 1$ denote by

$$c = \varphi(g(x)) = (g_0, \ldots, g_{n-k}, 0 \ldots, 0),$$
$$c' = \varphi(\tilde{h}(x)) = (h_k, \ldots, h_0, 0 \ldots, 0).$$

By Theorem 8.16, we know that $\mathcal{C}$ is generated by $c, \sigma(c), \ldots, \sigma^{k-1}(c)$ and $\mathcal{C}'$ is generated by $c', \sigma(c'), \ldots, \sigma^{n-k-1}(c')$.

It is enough to show that any element of the basis of $\mathcal{C}$, say $\sigma^i(c)$, is orthogonal to a basis of $\mathcal{C}'$, say $\sigma^j(c')$.

Let us assume that $0 \leq i < k, 0 \leq j < n - k$ and $j \geq i$ (the case $j < i$ works similarly).

Then,

$$\begin{aligned}\langle \sigma^i(c), \sigma^j(c')\rangle &= \langle c, \sigma^{j-i}(c')\rangle \\ &= g_{j-i}h_k + g_{j-i+1}h_{k-1} + \cdots + g_{j-i+k}h_0,\end{aligned}$$

where we set $g_\ell = 0$ if $\ell < k$.

We thus get the coefficient of order $j - i + k$ of their product $g(x)h(x)$. Since this is $x^n - 1$ and $0 < j - i + k < n$, we get that $\langle \sigma^i(c), \sigma^j(c')\rangle = 0$. □

Hence given a generator polynomial $g(x)$ of $\mathcal{C}$, we may compute

$$h(x) = (x^n - 1)/g(x)$$

and then get $\mathcal{C}^\perp$ is generated by $\tilde{h}(x)$.

**Example 8.27.** *In our previous example for $q = 2, n = 7$, and $g(x) = x^3 + x + 1$ we get that $h(x) = x^4 + x^2 + x + 1$ and hence $\tilde{h}(x) = x^4 + x^2 + 1$.*

*The generator matrix of the dual code is thus given by*

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

**Exercise 8.28.** *Give the generator polynomial of the primitive Reed-Solomon code $\mathcal{RS}_{q,n,k}(\alpha)$.*

**Exercise 8.29.** *Let us consider the code $\mathcal{C}$ over $\mathbb{F}_3$ generated by*

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

1. *Show that $\mathcal{C}$ is cyclic.*
2. *Find the generator polynomial of $\mathcal{C}$.*
3. *Find the generator polynomial of $\mathcal{C}^\perp$.*

## 8.3 Cyclotomic Classes

Let us focus on lengths $n$, which are coprime to $\text{char}(\mathbb{F}_q)$. Hence $x^n - 1$ has roots of multiplicity 1 in an extension field $\mathbb{F}_q^m$.

**Definition 8.30.** Let $n$ be coprime to $\text{char}(\mathbb{F}_q)$, then if $x^n - 1 = (x - a_1) \cdots (x - a_n)$ with $a_i \in \mathbb{F}_{q^m}$, (and $\mathbb{F}_{q^m}$ is the first extension field where this happens) then $\mathbb{F}_{q^m}$ is called *cyclotomic extension* of $\mathbb{F}_q$.

That is $\mathbb{F}_{q^m}$ is the smallest extension field containing all $n$-th roots of unity. In fact, if $a_i$ are roots of $x^n - 1$, then $a_i^n = 1$.

In particular to have elements of order $n$, we need $n \mid q^m - 1$.

Let us denote the $n$ $n$-th roots of unity by $1, \zeta_n, \ldots, \zeta_n^{n-1}$.

**Exercise 8.31.** *If $\zeta_n$ is an $n$-th root of unity, show that for all $i \in \{0, \ldots, n-1\}$ also $\zeta_n^i$ is an $n$-th root of unity.*

Clearly, since $\zeta_n^0 = \zeta_n^n = 1$, we should always consider the exponents of $\zeta_n$ modulo $n$.

Recall that we are interested in the irreducible factors of $x^n - 1$. Clearly over $\mathbb{F}_{q^m}$ we have

$$x^n - 1 = \prod_{i \in \mathbb{Z}/n\mathbb{Z}} (x - \zeta_n^i),$$

however $x - \zeta_n^i$ does not live in $\mathbb{F}_q[x]$.

**Lemma 8.32.** *Let $a_1, \ldots, a_m \in \mathbb{F}_{q^\ell}$ be distinct. Then $p(x) = \prod_{i=1}^m (x - a_i) \in \mathbb{F}_q[x]$ if and only if $a_i^q \in \{a_1, \ldots, a_m\}$ for all $i \in \{1, \ldots, m\}$.*

*Proof.* If $p(x) = \sum_{i=0}^m p_i x^i \in \mathbb{F}_q[x]$, then $p_i \in \mathbb{F}_q$ and hence $p_i^q = p_i$.

Thus

$$p(x^q) = \sum_{i=0}^m p_i (x^q)^i = \sum_{i=0}^m p_i^q (x^q)^i = \sum_{i=0}^m (p_i x^i)^q = (\sum_{i=0}^m p_i x^i)^q = (p(x))^q.$$

Thus if $a \in \mathbb{F}_{q^\ell}$ is such that $p(a) = 0 = p(a)^q = p(a^q)$, then $a^q$ is another root of $p(x)$. For the other direction we note (without proving it) that the sets of roots are closed under the Frobenius map.

□

Thus, in our case we are interested in the factors

$$g(x) = \prod_{i \in I} (x - \zeta_n^i),$$

for some $I \subseteq \mathbb{Z}/n\mathbb{Z}$. To have the condition from before, we require that for each $i \in I$ we have that

$$(\zeta_n^i)^q = \zeta_n^{iq} = \zeta_n^j$$

for some $j \in I$. Thus, the set $I \subseteq \mathbb{Z}/n\mathbb{Z}$ should be such that $qI = I$.

**Corollary 8.33.** *A factor of $x^n - 1$ in $\mathbb{F}_q[x]$ is of the form*

$$g(x) = \prod_{i \in I} (x - \zeta_n^i),$$

*where $I \subseteq \mathbb{Z}/n\mathbb{Z}$ is stable under multiplication with $q$.*

**Definition 8.34.** A *cylcotomic class* is a subset of $\mathbb{Z}/n\mathbb{Z}$ which is stable under multiplication with $q$. It is further called *minimal* if it is the smallest with respect to inclusion of cyclotomic classes.

We note that cyclotomic classes are in 1-to-1 correspondence with factors of $x^n - 1$ and futher minimal cyclotomic classes are in 1-to-1 correspondence with irreducible factors of $x^n - 1$. Thus

$$\{\text{cyclic codes over } \mathbb{F}_q^n\} \leftrightarrow \{\text{ideals in } \mathbb{F}_q[x]/(x^n-1)\} \leftrightarrow \{\text{monic factors of } x^n - 1\} \leftrightarrow \{\text{cyclotomic classes of } \mathbb{Z}/n\mathbb{Z}\}.$$

**Example 8.35.** *Let us consider again $q = 2, n = 7$, then the first extension field of $\mathbb{F}_2$ where we have 7th roots of unity is for $m = 3$. That is we are in $\mathbb{F}_8 = \mathbb{F}_2(\alpha)$ where $\alpha^3 = \alpha + 1$ and we can set $\zeta_7 = \alpha$.*

*We now compute the minimal cyclotomic classes of $\mathbb{Z}/7\mathbb{Z}$, as $\{0\}, \{1, 2, 4\}, \{3, 6, 5\}$.*
*Hence*

$$\begin{aligned}
\{0\} &\to (x + \alpha^0) = (x + 1), \\
\{1, 2, 4\} &\to (x + \alpha^1)(x + \alpha^2)(x + \alpha^4) = x^3 + x^2(\alpha + \alpha^2 + \alpha^4) \\
&\quad + x(\alpha\alpha^2 + \alpha\alpha^4 + \alpha^2\alpha^4) + \alpha\alpha^2\alpha^4 = x^3 + x + 1, \\
\{3, 6, 5\} &\to x^3 + x^2(\alpha^3 + \alpha^6 + \alpha^5) + x(\alpha^3\alpha^6 + \alpha^3\alpha^5 + \alpha^5\alpha^6) + \alpha^3\alpha^5\alpha^6 = x^3 + x^2 + 1.
\end{aligned}$$

*With this we have recovered that*

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

Some useful observations: for any cyclotomic class $A$, we have that $\prod_{i \in A} \zeta_n^i = 1$, for any $i, j \in \mathbb{Z}/n\mathbb{Z}$ we have that $\zeta_n^i \cdot \zeta_n^j = \zeta_n^{i+j}$.

**Exercise 8.36.** *Write out all cyclotomic classes of $\mathbb{Z}/7\mathbb{Z}$ and use them to compute all factors of $x^7 - 1 \in \mathbb{F}_2[x]$.*

## 8.4 Generalizations

Since we know how to compute the polynomial product $u(x) \cdot v(x) \in \mathbb{F}_q[x]/(x^n - 1)$, we can define a new vector multiplication in $\mathbb{F}_q^n$.

**Definition 8.37.** Let $u, v \in \mathbb{F}_q^n$ and define the *rotation matrix* as

$$\text{rot}(u) = \begin{pmatrix} u \\ \sigma(u) \\ \vdots \\ \sigma^{n-1}(u) \end{pmatrix}.$$

Let us denote by $u \circ v = u\text{rot}(v)$.

Observe that this is not much different to the multiplication matrix we have defined for the expansion map.

**Exercise 8.38.** *1. Show that* $\varphi(u \circ v) = u(x)v(x)$.

*2. Show that* $u \circ v = v \circ u$.

We may consider taking a different modulo. That is for $f(x) \in \mathbb{F}_q[x]$ of degree $n$, we can define new codes corresponding to ideals of $\mathbb{F}_q[x]/f(x)$.

We then again find a $\mathbb{F}_q$-vector space isomorphism

$$\psi : \mathbb{F}_q[x]/f(x) \to \mathbb{F}_q^n$$
$$a(x) = \sum_{i=0}^{n-1} a_i x^i \mapsto (a_0, \ldots, a_{n-1}).$$

While previously multiplication by $x$ corresponded to the right shift, we now have a different correspondence.

**Definition 8.39.** Let

$$\sigma : \mathbb{F}_q^n \to \mathbb{F}_q^n,$$
$$(a_0, \ldots, a_{n-1}) \mapsto \sigma(a) = \psi(xa(x)).$$

We again denote by $\sigma^i = \underbrace{\sigma \circ \cdots \circ \sigma}_{i \text{ times}}$.

**Exercise 8.40.** *Show that* $\sigma^i(a) = \psi(x^i a(x))$.

Hence, we also update our definition of a rotation matrix:

**Definition 8.41.** Let $f(x) \in \mathbb{F}_q[x]$ of degree $n$. The *ideal matrix* of $v \in \mathbb{F}_q^n$ is defined as

$$I_f(v) = \begin{pmatrix} v \\ \sigma(v) \\ \vdots \\ \sigma^{n-1}(v) \end{pmatrix}.$$

**Exercise 8.42.** *Let $u, v \in \mathbb{F}_q^n$. Show that*

$$u \circ v = uI_f(v) = I_f(v)^\top u = u \circ v.$$

*And $\psi(u(x)v(x)) = u \circ v$.*

We can then generalize the definition of a cyclic code:

**Definition 8.43.** Let $f(x) \in \mathbb{F}_q[x]$ of degree $n$. The *ideal code* $\mathcal{C}$ is then given by $\psi(\langle g(x) \rangle)$ for some $g(x) \in \mathbb{F}_q[x]/f(x)$ with $g(x) \mid f(x)$.

The generator matrix of an ideal code is similar to that of a cyclic code, simply using the ideal matrix instead of the rotation matrix:

**Exercise 8.44.** *Let $f(x) \in \mathbb{F}_q[x]$ of degree $n$. Let $\mathcal{C} = \psi(\langle g(x) \rangle)$ for some $g(x) \in \mathbb{F}_q[x]/f(x)$ with $deg(g) = n - k$. Show that*

$$G = \begin{pmatrix} g \\ \sigma(g) \\ \vdots \\ \sigma^{k-1}(g) \end{pmatrix}$$

**Example 8.45.** *Let us consider $\mathbb{F}_2$ and $f(x) = x^4 + x$. For $a(x) = \sum_{i=}^{3} a_i x^i \in \mathbb{F}_2[x]/f(x)$ we can compute*

$$\begin{aligned} xa(x) &= (a_3 + a_0)x + a_1x^2 + a_2x^3, \\ x^2a(x) &= a_2x + (a_3 + a_0)x^2 + a_1x^3, \\ x^3a(x) &= a_1x + a_2x^2 + (a_3 + a_0)x^3. \end{aligned}$$

*Hence we get*

$$\begin{aligned} \sigma(a_0, a_1, a_2, a_3) &= (0, a_3 + a_0, a_1, a_2), \\ \sigma^2(a_0, a_1, a_2, a_3) &= (0, a_2, a_3 + a_0, a_1), \\ \sigma^3(a_0, a_1, a_2, a_3) &= (0, a_1, a_2, a_3 + a_0). \end{aligned}$$

*Let $g(x) = (x^2 + x) \in \mathbb{F}_2[x]/(x^4 + x)$, then*

$$\langle g(x) \rangle = \{0, x^2 + x, x^3 + x^2, x^3 + x\}$$

*and*

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

Finally, we introduce quasi-cyclic codes. For $a = (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$ and some $\ell \in \{1, \ldots, n\}$ we denote by $\sigma_\ell(x)$ its *$\ell$-cyclic shift*, i.e.,

$$\sigma_\ell(x) = (x_{0+\ell}, \ldots, x_{n-1+\ell}),$$

where the indices $i + \ell$ should be considered modulo $n$.

**Definition 8.46.** An $[n, k]_q$ linear code $\mathcal{C}$ is a *quasi-cyclic* code, if there exists $\ell \in \mathbb{N}$, such that $\sigma_\ell(\mathcal{C}) = \mathcal{C}$.

In addition, if $n = \ell a$, for some $a \in \mathbb{N}$, then it is convenient to write the generator matrix composed into $a \times a$ circulant matrices.

Let $\mathcal{C}$ be a $[n, k]_q$ linear code.

- $\mathcal{C}$ is *cyclic* if $\sigma(\mathcal{C}) = \mathcal{C}$.
- $\mathcal{C}$ is isomorphic to $\langle g(x) \rangle \subseteq \mathbb{F}_q[x]/(x^n - 1)$ for $g(x) \mid (x^n - 1)$ of degree $\deg(g) = n - k$.
- $g(x)$ is called the *generator polynomial* of $\mathcal{C}$.
- $\mathcal{C}^\perp$ is cyclic with generator polynomial $\tilde{h}(x)$, where $h(x) = (x^n - 1)/g(x)$.
- The finite field containing all $n$ $n$-th roots of unity $1 = \zeta_n^0, \zeta_n, \ldots, \zeta_n^{n-1}$, is called *cyclotomic extension field.*
- $I \subseteq \mathbb{Z}/n\mathbb{Z}$ with $qI = I$ is called *cyclotomic class.*
- $g(x) = \prod_{i \in I}(x - \zeta_n^i) \in \mathbb{F}_q[x]$ if and only if $I$ is a cyclotomic class.
- $\mathcal{C}$ is called *ideal code*, if it corresponds to an ideal of $\mathbb{F}_q[x]/f(x)$.

# 9 Generic Decoding

We have seen an efficient decoder for GRS codes and to ensure reliable communication is also efficient, this is a condition we impose on families of codes.

However, if we have no information on the algebraic structure of the code, or even try to decode a random code, a new question arises: how hard is it to decode in general?

This will be the main task of this chapter: *generic decoding*.

## 9.1 Interlude: Code-based Cryptography

Such generic decoders have a great impact for cryptography: we have hinted at it several times (and next semester there will be a lecture purely on this subject): coding theory can also be used in cryptography. This intersection is called *code-based cryptography* and is as old as the RSA cryptosystem (i.e., from 1978 [13]).

On a very high level, the idea of the McEliece public-key encryption scheme is to use a code with an efficient decoder (he suggested Goppa codes) as a secret key, and to publish an equivalent code, where the structure of the secret code is hidden (and thus also the necessary information for the decoder). That is, we take as secret key $G \in \mathbb{F}_q^{k\times n}$, a generator matrix for the secret code, and publish $G' = SGP$ for some $S \in \mathrm{GL}_q(k), P$ a permutation matrix, and the error-correction capability $t$.

Anyone can then encrypt a message $m \in \mathbb{F}_q^k$, by computing a corrupted codeword:

$$c = mG' + e,$$

where $\mathrm{wt}_H(e) = t$.

The person with the secret key, i.e., $S, P$, and the decoding algorithm of $\mathcal{C}$, can then compute

$$cP^\top = mSG + eP^\top = m'G + e',$$

and since $\mathrm{wt}_H(eP^\top) = \mathrm{wt}_H(e') = \mathrm{wt}_H(e) = t$, the decoder for $\mathcal{C} = \langle G\rangle$, will return $m' = mS$. Finally by multiplying with $S^{-1}$, we recover the message $m$.

An eavesdropper has only access to the public key, i.e., $G'$ and the corrupt codeword $c$. Thus their main task, is to decode a (seemingly) random code.

Clearly, in this scenario, the public code is not random: it is still an equivalent code to a Goppa code. By now, we also have new code-based cryptosystems, where we employ actually random codes. If you want to learn more about these cryptosystems and their security, sign up for the lecture "Code-Based Cryptography".

## 9.2 Decoding Problem

The main problem of this chapter, is called *Decoding Problem (DP).*

**Problem 9.1** (Decoding Problem)**.** *Let $\mathbb{F}_q$ be a finite field and $k \leq n$ be positive integers. Given $G \in \mathbb{F}_q^{k\times n}$, $r \in \mathbb{F}_q^n$ and $t \in \mathbb{N}$, find $e \in \mathbb{F}_q^n$ with $wt_H(e) = t$ and $r - e \in \langle G \rangle$.*

Clearly, generic decoders are algorithms that solve the DP.

Our first question, "how hard is it to decode in general?" has already been solved: The DP has been proven to be NP-hard [3], meaning it is one of the hardest problems in mathematics.

The problem is further also in NP. This means, if we are given a candidate solution we can easily (in polynomial time) check whether it is actually a solution. Thus, the DP is a NP-complete problem, which makes it a perfect candidate for cryptography.

Note that the DP is formulated through the generator matrix and we can get an equivalent formulation using the parity-check matrix: the *Syndrome Decoding Problem (SDP).*

**Problem 9.2** (Syndrome Decoding Problem)**.** *Let $\mathbb{F}_q$ be a finite field and $k \leq n$ be positive integers. Given $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$ and $t \in \mathbb{N}$,find $e \in \mathbb{F}_q^n$ such that $wt_H(e) = t$ and $eH^\top = s$.*

These two problems are also equivalent to the *Codeword Finding Problem (CFP)*:

**Problem 9.3** (Codeword Finding Problem)**.** *Let $\mathbb{F}_q$ be a finite field and $k \leq n$ be positive integers. Let $k \leq n$ be positive integers. Given $H \in \mathbb{F}_q^{(n-k)\times n}$ and $t \in \mathbb{N}$, find $c \in \mathbb{F}_q^n$ such that $wt_H(c) = t$ and $cH^\top = 0$.*

**Theorem 9.4.** *The DP, SDP and CFP are equivalent.*

*Proof.* Let us start with showing that the DP and SDP are equivalent. For this we start with an instance of DP, i.e., $G, r, t$. We can then transform this instance to an instance of the SDP. In fact, we can bring $G$ into systematic form, that is $\begin{pmatrix}\mathrm{Id}_k & A\end{pmatrix}$ and immediately get a parity-check matrix for the same code $H = \begin{pmatrix}-A^\top & \mathrm{Id}_{n-k}\end{pmatrix}$.

We can then multiply $H$ to the received vector $r = mG + e$, getting the syndrome

$$s = rH^\top = eH^\top.$$

Hence, if we can solve the SDP on the instance $H, s, t$, that is we find $e$, we have also solved the DP.

On the other hand, given an instance of SDP, i.e., $H, s, t$, we can find an instance of DP: we bring $H$ into systematic form $\begin{pmatrix}B & \mathrm{Id}_{n-k}\end{pmatrix}$ and read of a generator matrix $G = \begin{pmatrix}\mathrm{Id}_k & -B^\top\end{pmatrix}$ for the same code.

We can now solve

$$xH^\top = s \tag{2}$$

for some unknown $x \in \mathbb{F}_q^n$ and since this is a linear system of $n-k$ equations in $n$ unknowns, we get $N = q^k$ possible solutions: $x_1, \ldots, x_N$. Note that for each of the $q^k$ codewords $c_1, \ldots, c_N$, we have that $c_i + e$ is a possible solution to (2). Thus, each of the $q^k$ solutions $x_i$ correspond to some $c_i + e$.

Hence, any of the solutions $x_i$ can be used as received vector $r$ and we have recovered an instance of DP, as $G, r, t$. Hence, solving DP, i.e., finding $e$, also solves the SDP instance.

Finally, we show that the DP and SDP are also equivalent to CFP. For this, we recall that the error-correction capability $t$ is set as $t = \lfloor \frac{d-1}{2} \rfloor$, where $d$ denotes the minimum distance of $\langle G \rangle$.

Given an instance of DP, i.e., $G, r, t$ we can add $r$ as a row to the generator matrix, getting

$$G' = \begin{pmatrix} G \\ r \end{pmatrix}.$$

Note that the code generated by $G'$ is also generated by

$$\begin{pmatrix} G \\ e \end{pmatrix},$$

as $r = mG + e$. The new code of dimension $k+1$ has now as lowest weight codeword $\lambda e$ of weight $t$ for some $\lambda \in \mathbb{F}_q^\times$.

In fact, if there would exist some codeword $a \in \langle G' \rangle$ of weight $t$, which is not of the form $a = \lambda e$, then $a$ must also involve codewords of $\langle G \rangle$, that is $a = c + be$, for $c \in \langle G \rangle \setminus \{0\}$ and some $b \in \mathbb{F}_q$.

Since we know $\mathrm{wt}_H(e) = t$, we must have that $c = a - be \in \langle G \rangle \setminus \{0\}$ has weight

$$\mathrm{wt}_H(c) = \mathrm{wt}_H(a - be) \leq \mathrm{wt}_H(a) + \mathrm{wt}_H(-be) = 2t < d,$$

a contradiction to the minimum distance of $\langle G \rangle$ being $d$.

Hence, we can compute the corresponding parity-check matrix $H'$ of $\langle G' \rangle$ and solving the CFP on the instance $H', t$ we recover the solution $e$ to the DP instance.

On the other hand, given an instance $H, t$ of the CFP, we can define an instance of SDP, by taking the same parity-check matrix and setting the syndrome $s = 0$. Thus, a solver for SDP, searching for a weight $t$ vector $e$ with $eH^\top = 0$ also solves the CFP instance. □

Thus, we may choose which of the three problems we wish to solve with our generic decoder. For this lecture, we will stick to the SDP.

## 9.3 Solvers

Recall that the SDP is given $H \in \mathbb{F}_q^{(n-k)\times n}, s \in \mathbb{F}_q^{n-k}, t \in \mathbb{N}$ and searches for a vector $e \in \mathbb{F}_q^n$ with $eH^\top = s$ and $\mathrm{wt}_H(e) = t$.

For our solvers, we will restrict ourselves to weights $t$ up to the correction capability of the code, that is $t \leq \lfloor \frac{d-1}{2} \rfloor$, where $d$ denotes the minimum distance of the code $\ker(H^\top)$, and we will assume that a solution $e$ exists. Thus, it is also the unique solution.

**Exercise 9.5.** *Show that if $e, e' \in \mathbb{F}_q^n$ are both solutions to the SDP with $t \leq \lfloor \frac{d-1}{2} \rfloor$, then $e = e'$.*

Note that any generic decoding algorithm will not "break" a code-based cryptosystem: these algorithms have an exponential cost (in $n$)! Instead, we use them to determine which parameters $(q, n, k, t)$ we should use to reach a given security level, e.g. $\lambda = 128$ bits.

In the following, we will often simply say "an object is random", meaning it is chosen uniform at random from the whole space.

Since we assume that $\mathcal{C} = \ker(H^\top)$ in the SDP is random, let us quickly recall what we have learned so far about random codes: we may assume that $H \in \mathbb{F}_q^{(n-k)\times n}$ has full rank and no zero column, thus there exists some invertible matrix $U$ and a permutation matrix $P$, such that

$$UHP = \begin{pmatrix} \mathrm{Id}_{n-k} & A \end{pmatrix},$$

where $A \in \mathbb{F}_q^{(n-k)\times k}$ is a random matrix with no zero column.

- For large $n$, we can assume their minimum distance of $\mathcal{C}$ is given by the GV bound. Asymptotically this lets us set $\delta = d/n$ is

  $$\delta = H_q^{-1}(1 - R),$$

  where $R = k/n$.

- The probability for a random set $J \subset \{1, \ldots, n\}$ of size $k$ to be an information set is large and ignored in the cost computations:

  $$\frac{\prod_{i=0}^{k-1}(q^k - q^i)}{q^{k^2}} = \prod_{i=1}^{k}(1 - q^{-i}).$$

- A random vector of $\mathbb{F}_q^n$ has probability $q^{-(n-k)}$ to be in the code.

- The syndrome of a random vector $x \in \mathbb{F}_q^n$ is also random.

Let us start with some remarks on what makes the SDP so hard to solve:

The two conditions on $e$ of the SDP are not compatible:

1. the parity-check equation $eH^\top = s$, is a linear constraint, while
2. the weight constraint $\mathrm{wt}_H(e) = t$ is non-linear.

One condition alone is clearly not hard to solve: the first one is a linear system with $n-k$ equations and $n$ unknowns, for which we can find a solution in polynomial time. We can also simply list all vectors of weight $t$.

However, it is very unlikely that any solution for one of the conditions will also satisfy the other. In particular, since we assumed that in their intersection, we only have a unique solution.

The first try we could have at solving the DP is straightforward: solve only one of the conditions and check for the other. These are the two brute-force algorithms.

**Algorithm 1** Brute-Force Decoding 1

Input: $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{N}$.
Output: $e \in \mathbb{F}_q^n$ with $eH^\top = s$ and $\mathrm{wt}_H(e) = t$.

1: Find the solution set $\mathcal{L}$ to the linear system $xH^\top = s$.
2: **for** $x \in \mathcal{L}$ **do**
3: **if** $\mathrm{wt}_H(x) = t$ **then**
4: Return $x$

To estimate the cost of algorithms, we use the big-O notation.

**Definition 9.6.** Let $f(x), g(x)$ be functions over the reals. We write $f(x) \in \mathcal{O}(g(x))$ to denote that there exists a positive real constant $N$ such that $|f(x)| < Ng(x)$ for all $x > x_0$, meaning that if $x$ grows, $f(x)$ will not grow faster than $g(x)$.

**Proposition 9.7.** *The Brute-Force Algorithm 1 has a cost in $\mathcal{O}(q^k)$.*

*Proof.* This follows easily from the observation that the solution set $\mathcal{L}$ is expected to have size $q^k$. In fact, any vector $x \in \mathbb{F}_q^n$ has probability $q^{-(n-k)}$ to have syndrome $s$, thus in total there are $q^n q^{-(n-k)} = q^k$ many vectors with syndrome $s$.

Out of all these vectors only one has weight $t$, thus the cost of such brute-force algorithm is in

$$\mathcal{O}(|\mathcal{L}|) = \mathcal{O}(q^k).$$

□

Similarly, we can go through the vectors of weight $t$ and check if the syndrome equations are satisfied. For this we denote by

$$S_H(t, n, q) = \{x \in \mathbb{F}_q^n \mid \mathrm{wt}_H(x) = t\}$$

the Hamming sphere of radius $t$.

**Algorithm 2** Brute-Force Decoding 2

Input: $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{N}$.
Output: $e \in \mathbb{F}_q^n$ with $eH^\top = s$ and $\mathrm{wt}_H(e) = t$.

1: Build the list $S_H(t, n, q)$ of all vectors of weight $t$.
2: **for** $x \in S_H(t, n, q)$ **do**
3: **if** $xH^\top = s$ **then**
4: Return $x$

**Proposition 9.8.** *The Brute-Force Algorithm 2 has a cost in* $\mathcal{O}\left(\binom{n}{t}(q-1)^t\right)$.

*Proof.* Note that

$$S_H(t,n,q) = \binom{n}{t}(q-1)^t$$

and since there exists a unique solution to the SDP, we are expected to go through all elements of $S_H(t,n,q)$. □

## 9.4 ISD Algorithms

There are of course a more clever algorithms to solve SDP than brute-forcing, but we note that their costs will remain exponential.

The main idea is to use the information sets. These types of algorithms have been initialized by Prange in 1962 [14], and are called *Information Set Decoding (ISD)* algorithms.

As they all follow a similar structure, we will first introduce them on a high-level: We start by picking an information set $I$ and assume a certain weight of $e_I$, say $w$, and thus impose $e_{I^C}$ has the remaining weight $t-w$.

By doing so, we may solve a smaller problem than the initial SDP instance.

Let $H \in \mathbb{F}_q^{(n-k)\times n}$, and $I$ be an information set, then there exists $U \in \mathrm{GL}_q(n-k)$ and $P$ a permutation matrix such that

$$UHP = H' = \begin{pmatrix}\mathrm{Id}_{n-k} & A\end{pmatrix},$$

for some $A \in \mathbb{F}_q^{(n-k)\times k}$. Since $eH^\top = s$ we now have

$$e'H'^\top = (eP)(P^\top H^\top U^\top) = sU^\top = s',$$

and hence we can rewrite the parity-check equation

$$\begin{pmatrix}\mathrm{Id}_{n-k} & A\end{pmatrix}\begin{pmatrix}e_{I^C}^\top \\ e_I^\top\end{pmatrix} = s'^\top$$

as

$$e_{I^C} + e_I A^\top = s'.$$

Thus if we find $e_I \in \mathbb{F}_q^k$, of weight $w$ and such that $e_I A^\top = \tilde{s}$, for some $\tilde{s} \in \mathbb{F}_q^{n-k}$, then we are left with checking that

$$\mathrm{wt}_H(e_{I^C}) = \mathrm{wt}_H(s' - \tilde{s}) = t - w.$$

We can generalize this idea further, as the new ISD algorithms actually find a set $J \subseteq \{1,\dots,n\}$ of size $k+\ell$, for some $\ell \leq n-k$, containing an information set $I$.

Then, we bring $H$ into a *quasi-systematic form*, i.e.,

$$H' = \begin{pmatrix}\mathrm{Id}_{n-k-\ell} & A \\ 0 & B\end{pmatrix},$$

where $A \in \mathbb{F}_q^{(n-k-\ell)\times(k+\ell)}$, $B \in \mathbb{F}_q^{\ell\times(k+\ell)}$. We then also split the syndrome $s'$ accordingly: $s' = \begin{pmatrix} s_1 & s_2 \end{pmatrix}$, where $s_1 \in \mathbb{F}_q^{n-k-\ell}$ and $s_2 \in \mathbb{F}_q^{\ell}$.

The parity-check equation $e'H'^{\top} = s'$ becomes two equations:

$$e_{J^C} + e_J A^{\top} = s_1, \tag{3}$$

$$e_J B^{\top} = s_2. \tag{4}$$

Note that if we find a solution $e_J \in \mathbb{F}_q^{k+\ell}$ of weight $w$, to (4), we can simply check if

$$\mathrm{wt}_H(e_{J^C}) = \mathrm{wt}_H(s_1 - e_J A^{\top}) = t - w.$$

Thus, we have reduced the initial SDP with instance $(H, s, t)$ into a smaller SDP with instance $(B, s_2, w)$.

We may summarize the general idea of ISD as:

1. Find a set $J \subset \{1, \ldots, n\}$ of size $k + \ell$ containing an information set for $\mathcal{C}$.
2. Find an invertible matrix $U \in \mathbb{F}_q^{(n-k)\times(n-k)}$, and a permutation matrix $P$, such that
$$UHP = \begin{pmatrix} \mathrm{Id}_{n-k-\ell} & A \\ 0 & B \end{pmatrix}.$$
3. Compute $s' = sU^{\top}$ and split it into $s_1, s_2$.
4. Find $e_J \in \mathbb{F}_q^{k+\ell}$ of weight $w$ and such that $e_J B^{\top} = s_2$.
5. Check if $\mathrm{wt}_H(e_{J^C}) = \mathrm{wt}_H(s_1 - e_J A^{\top}) = t - w$.
6. If this is satisfied, output $e = (e_J, e_{J^C})P^{\top}$, if not start over with a new choice of $J$.

Note that for a fixed set $J$, the sought error vector $e$ might not be such that $e_J$ has weight $w$. Thus, the iteration above has to be repeated several times, and the final cost of such algorithm is given by the cost of one iteration times the expected number of required iterations.

On average, the number of iterations required is given by the reciprocal of the success probability of one iteration and this probability is completely determined by the assumed weight distribution.

**Lemma 9.9.** *Let $k \leq n$ and $w \leq t$ be positive integers. Let $e \in \mathbb{F}_q^n$ be of weight $t$. For a randomly chosen $J \subset \{1, \ldots, n\}$ of size $k + \ell$, the probability that $\mathrm{wt}_H(e_J) = w$ is given by*

$$\binom{t}{w}\binom{n-t}{k+\ell-w}\binom{n}{k+\ell}^{-1}.$$

We note that fixing $e$ and going through all possible choices of $J$, is indeed what the algorithm tells us to do. However, to compute the success probability of one iteration, it is usually easier to go the other direction: fix a set $J$ and compute the probability that $e$ has the desired weight distribution.

**Lemma 9.10.** *Let $k \leq n$ and $w \leq t$ be positive integers. Let $J \subset \{1, \ldots, n\}$ be of size $k + \ell$. For a randomly chosen $e \in \mathbb{F}_q^n$ of weight $t$, the probability that $\mathrm{wt}_H(e_J) = w$ is given by*

$$\binom{k+\ell}{w}\binom{n-k-\ell}{t-w}\binom{n}{t}^{-1}.$$

**Exercise 9.11.** *Prove Lemma 9.9 and 9.9 and show that the two probabilities are the same.*

### 9.4.1 Prange's Algorithm

In Prange's algorithm we assume that there exists an information set $I$ that is disjoint to the support of the error vector $\mathrm{supp}_H(e)$, i.e.,

$$I \cap \mathrm{supp}_H(e) = \emptyset.$$

Thus in terms of our previous general algorithm for ISD, we set $w = \ell = 0$ and hence $J = I$ and $e_I = 0$.

To illustrate the algorithm, let us assume that the information set is $I = \{1, \ldots, k\}$. To bring the parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ into systematic form, we multiply by an invertible matrix $U \in \mathbb{F}_q^{(n-k)\times(n-k)}$. Since we assume that no errors occur in the information set, we have that $e = (0, e_{I^C})$ with $\mathrm{wt}_H(e_{I^C}) = t$. We are in the following situation:

$$UHe^\top = \begin{pmatrix}\mathrm{Id}_{n-k} & A\end{pmatrix}\begin{pmatrix}e_{I^C}^\top \\ 0^\top\end{pmatrix} = Us^\top,$$

for $A \in \mathbb{F}_q^{(n-k)\times k}$.

It follows that $e_{I^C} = sU^\top$ and hence we are only left with checking the weight of $s' = sU^\top$.

**Algorithm 3** Prange's Algorithm

Input: $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{N}$.
Output: $e \in \mathbb{F}_q^n$ with $eH^\top = s$ and $\mathrm{wt}_H(e) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size $k$.
2: Compute $U \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that

$$(UH)_I = A \text{ and } (UH)_{I^C} = \mathrm{Id}_{n-k},$$

where $A \in \mathbb{F}_q^{(n-k)\times k}$.
3: Compute $s' = sU^\top$.
4: **if** $\mathrm{wt}_H(s') = t$ **then**
5: Return $e$ such that $e_I = 0$ and $e_{I^C} = s'$.
6: Start over with Step 1 and a new selection of $I$.

**Theorem 9.12.** *Prange's algorithm has a cost in*

$$\mathcal{O}\left(\binom{n-k}{t}^{-1}\binom{n}{t}\right).$$

*binary operations.*

*Proof.* One iteration of Algorithm 3 only consists of bringing $H$ into systematic form and applying the same row operations on the syndrome; thus, the cost can be assumed equal to that of computing $U\begin{pmatrix} H & s^\top \end{pmatrix}$, i.e., $(n-k)^2(n+1)$ $\mathbb{F}_q$-operations or

$$(n-k)^2(n+1)(\lceil \log_2(q) \rceil + \lceil \log_2(q) \rceil^2)$$

binary operations.

The success probability is given by having chosen the correct weight distribution of $e$. In this case, we require that no errors happen in the chosen information set, hence the probability is given by

$$\binom{n-k}{t}\binom{n}{t}^{-1}.$$

Since the average number of iterations are then $\binom{n-k}{t}^{-1}\binom{n}{t}$ and

$$(n-k)^2(n+1)\left(\lceil \log_2(q) \rceil + \lceil \log_2(q) \rceil^2\right)\binom{n-k}{t}^{-1}\binom{n}{t} \in \mathcal{O}\left(\binom{n-k}{t}^{-1}\binom{n}{t}\right),$$

the Gaussian elimination part introduces only polynomial factors which we may ignore. □

Let us consider an example for Prange's algorithm.

**Example 9.13.** *Let us consider* $\mathbb{F}_5$ *and*

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 0 & 1 & 3 & 4 \end{pmatrix}, \qquad s = (2, 4, 1)$$

*and we are looking for* $e \in \mathbb{F}_5^3$ *with weight* $t = 1$.

*We might start with the information set* $I_1 = \{4, 5\}$ *as* $H$ *is already in systematic form for* $I_1$. *That is the necessary* $U_1 = Id_3$, *however* $s' = s$ *does not have weight* $t = 1$.

*Instead, the information set* $I_2 = \{1, 2\}$ *leads to*

$$U_2 = \begin{pmatrix} 1 & 3 & 1 \\ 2 & 2 & 0 \\ 2 & 4 & 0 \end{pmatrix},$$

*i.e.,*

$$U_2 H = \begin{pmatrix} 1 & 3 & 1 & 0 & 0 \\ 2 & 2 & 0 & 1 & 0 \\ 2 & 4 & 0 & 0 & 1 \end{pmatrix}.$$

*Now we get* $s' = sU_1\top = (0, 2, 0)$ *has weight 1, and hence we found the error vector*

$$e = (0, 0, 0, 2, 0).$$

## 9.5 Stern's Algorithm

A few years later in 1988, Stern proposed a meet-in-the-middle approach to solve for the smaller instance.

Recall that when we are given the instance $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$ and $t$ and are searching for $e \in \mathbb{F}_q^n$ such that $\mathrm{wt}_H(e) = t$ and $eH^\top = s$, we may first reduce it to a smaller instance, $e' \in \mathbb{F}_q^{k+\ell}$ with $\mathrm{wt}(e') = w$ and $e'H'^\top = s'$, where $H' \in \mathbb{F}_q^{\ell\times(k+\ell)}$, $s' \in \mathbb{F}_q^\ell$.

Stern proposes to split $e' = (e_1, e_2)$, where $e_i \in \mathbb{F}_q^{(k+\ell)/2}$ are of weight $w/2$ and similarly to split $H' = \begin{pmatrix} H_1 & H_2 \end{pmatrix}$, with $H_i \in \mathbb{F}_q^{\ell\times(k+\ell)/2}$. The syndrome equation $e'H\top = s'$ then becomes

$$\begin{pmatrix} H_1 & H_2 \end{pmatrix} \begin{pmatrix} e_1^\top \\ e_2^\top \end{pmatrix} = s'^\top$$

that is

$$e_1 H_1^\top + e_2 H_2^\top = s'.$$

If we denote (for the correct choice $e$) that $e_1 H_1^\top = s_1$ and $e_2 H_2^\top = s_2$, then it is again enough to find a pair $(e_1, e_2)$ such that $s_1 + s_2 = s'$. Thus we simply set $s_1 = s' - e_2 H_2^\top$.

We may then build two lists

$$\mathcal{L}_1 = \{(e_1 H_1^\top, e_1) \mid e_1 \in \mathbb{F}_q^{(k+\ell)/2}, \mathrm{wt}_H(e_1) = w/2\},$$
$$\mathcal{L}_2 = \{(s' - e_2 H_2^\top, e_2) \mid e_2 \in \mathbb{F}_q^{(k+\ell)/2}, \mathrm{wt}_H(e_2) = w/2\}.$$

If we find a collision, that is $((a, e_1), (a, e_2)) \in \mathcal{L}_1 \times \mathcal{L}_2$, we know that $e_1 H_1^\top = s' - e_2 H_2^\top$ and hence $(e_1, e_2) H'^\top = s'$.

**Algorithm 4** Stern's Algorithm

Input: $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$, $w < t, \ell < n-k$.
Output: $e \in \mathbb{F}_q^n$ with $eH^\top = s$ and $\mathrm{wt}_H(e) = t$.

1: Choose a set $J \subset \{1, ..., n\}$ of size $k + \ell$.
2: Compute $U \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that

$$(UH)_J = \begin{pmatrix} A \\ H' \end{pmatrix}, \ (UH)_{J^C} = \begin{pmatrix} \mathrm{Id}_{n-k-\ell} \\ 0 \end{pmatrix},$$

where $A \in \mathbb{F}_q^{(n-k-\ell)\times(k+\ell)}$ and $H' \in \mathbb{F}_q^{\ell\times(k+\ell)}$.
3: Split $H' = (H_1, H_2)$, with $H_i \in \mathbb{F}_q^{\ell\times(k+\ell)/2}$.
4: Compute $sU^\top = \begin{pmatrix} \tilde{s} & s' \end{pmatrix}$, where $\tilde{s} \in \mathbb{F}_q^{n-k-\ell}$ and $s' \in \mathbb{F}_q^\ell$.
5: Compute the sets

$$\mathcal{L}_1 = \{(e_1 H_1^\top, e_1) \mid e_1 \in \mathbb{F}_q^{(k+\ell)/2}, \mathrm{wt}_H(e_1) = w/2\},$$
$$\mathcal{L}_2 = \{(s' - e_2 H_2^\top, e_2) \mid e_2 \in \mathbb{F}_q^{(k+\ell)/2}, \mathrm{wt}_H(e_2) = w/2\}.$$

6: **for** $(a, e_1) \in \mathcal{L}_1$ **do**
7: **for** $(a, e_2) \in \mathcal{L}_2$ **do**
8: **if** $\mathrm{wt}_H(\tilde{s} - (e_1, e_2)A^\top) = t - w$ **then**
9: Return $e$ such that $e_J = (e_1, e_2)$, $e_{J^C} = \tilde{s} - (e_1, e_2)H'^\top$.
10: Start over with Step 1 and a new selection of $J$.

**Theorem 9.14.** *Stern's algorithm has a cost in*

$$\mathcal{O}\left(\binom{(k+\ell)/2}{w/2}^{-2}\binom{n-k-\ell}{t-w}^{-1}\binom{n}{t}\left(\binom{(k+\ell)/2}{w/2}(q-1)^{w/2} + \binom{(k+\ell)/2}{w/2}^2(q-1)^{w-\ell}\right)\right).$$

*Proof.* We start with the cost of one iteration. Again, the computation of $UH$ is only polynomial in $n$ and thus negligible. On the other hand, the construction of the lists $\mathcal{L}_i$ costs

$$|\mathcal{L}_i| = \binom{(k+\ell)/2}{w/2}(q-1)^{w/2}.$$

To go through $\mathcal{L}_1 \times \mathcal{L}_2$ would usually cost $|\mathcal{L}_i|^2$, but since we are only interested in collisions, i.e., when $s' - e_2 H_2^\top = e_1 H_1^\top \in \mathbb{F}_q^\ell$, we can multiply $|\mathcal{L}_i|^2$ with the probability of having a collision, that is $q^{-\ell}$.

We get that the cost of one iteration is in

$$\mathcal{O}\left(\binom{(k+\ell)/2}{w/2}(q-1)^{w/2} + \binom{(k*\ell)/2}{w/2}^2 (q-1)^{w-\ell}\right).$$

For the success probability of one iteration, we need to compute

$$\frac{|\{e \in \mathbb{F}_q^n \mid e_J = (e_1, e_2), \mathrm{wt}_H(e_i) = w/2, \mathrm{wt}_H(e_{J^C}) = t - w\}|}{|\{e \in \mathbb{F}_q^n \mid \mathrm{wt}_H(e) = t\}},$$

which is given by

$$\binom{(k+\ell)/2}{w/2}^2 \binom{n-k-\ell}{t-w} \binom{n}{t}^{-1}.$$

Thus the overall cost of Stern's algorithm is in

$$\mathcal{O}\left(\binom{(k+\ell)/2}{w/2}^{-2} \binom{n-k-\ell}{t-w}^{-1} \binom{n}{t} \left(\binom{(k+\ell)/2}{w/2}(q-1)^{w/2} + \binom{(k+\ell)/2}{w/2}^2 (q-1)^{w-\ell}\right)\right).$$

□

**Example 9.15.** *Let us consider again $\mathbb{F}_5, n = 10, k = 4, t = 3, w = 2$, and $\ell = 2$.*

*Let*

$$H = \left(\begin{array}{cccc} & 1 & 2 & 3 & 1 \\ & 2 & 4 & 1 & 2 \\ & 3 & 3 & 2 & 4 \\ Id_6 & 1 & 2 & 1 & 3 \\ & 4 & 1 & 1 & 2 \\ & 3 & 3 & 4 & 1 \end{array}\right), \qquad s = (1, 0, 3, 1, 4, 4).$$

*If we set $J = \{5, 6, 7, 8, 9, 10\}$ then this clearly contains the information set $I = \{7, 8, 9, 10\}$, for which $H$ is already in systematic form.*

*Thus, we get*

$$H' = \begin{pmatrix} 1 & 0 & 4 & 1 & 1 & 2 \\ 0 & 1 & 3 & 3 & 4 & 1 \end{pmatrix}, \quad s' = (4, 4).$$

*We can then split $H'$ into $H_1 = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & 3 \end{pmatrix}, H_2 = \begin{pmatrix} 1 & 1 & 2 \\ 3 & 4 & 1 \end{pmatrix}$.*

*We build the lists*

$$\begin{aligned}
\mathcal{L}_1 &= \{(e_1 H_1^\top, e_1) \mid e_1 \in \mathbb{F}_5^3, wt_H(e_1) = 1\} \\
&= \{((\lambda, 0), (\lambda, 0, 0)), ((0, \lambda), (0, \lambda, 0)), ((4\lambda, 3\lambda), (0, 0, \lambda)) \mid \lambda \in \mathbb{F}_5^\times\}, \\
\mathcal{L}_2 &= \{(s' - e_2 H_2^\top, e_2) \mid e_2 \in \mathbb{F}_5^3, wt_H(e_2) = 1\} \\
&= \{((4 - \lambda, 4 - 3\lambda), (\lambda, 0, 0)), ((4 - \lambda, 4 - 4\lambda), (0, \lambda, 0)), ((4 - 2\lambda, 4 - \lambda), (0, 0, \lambda)) \mid \lambda \in \mathbb{F}_5^\times\}.
\end{aligned}$$

*Both lists have size* $12 = \binom{3}{1}(5-1)^1 = 3 \cdot 4$.

*We then search for collisions among the two lists, we find*

$$\begin{aligned}
&((1,0),(1,0,0)) \in \mathcal{L}_1, ((1,0),(3,0,0)) \in \mathcal{L}_2,\\
&((3,0),(3,0,0)) \in \mathcal{L}_1, ((3,0),(0,1,0)) \in \mathcal{L}_2,\\
&((1,0),(1,0,0)) \in \mathcal{L}_1, ((1,0),(0,0,4)) \in \mathcal{L}_2,\\
&((0,1),(0,1,0)) \in \mathcal{L}_1, ((0,1),(4,0,0)) \in \mathcal{L}_2,\\
&((0,3),(0,3,0)) \in \mathcal{L}_1, ((0,3),(0,4,0)) \in \mathcal{L}_2,\\
&((0,2),(0,2,0)) \in \mathcal{L}_1, ((0,2),(0,0,2)) \in \mathcal{L}_2,\\
&((3,1),(0,0,2)) \in \mathcal{L}_1, ((3,1),(1,0,0)) \in \mathcal{L}_2,\\
&((1,2),(0,0,4)) \in \mathcal{L}_1, ((1,2),(0,3,0)) \in \mathcal{L}_2,\\
&((3,1),(0,0,2)) \in \mathcal{L}_1, ((3,1),(0,0,3)) \in \mathcal{L}_2.
\end{aligned}$$

*Which are more than the expected* $|\mathcal{L}_i|^2 5^{-2} = 5.76$ *collisions.*

*For each of the candidate* $e' = (e_1, e_2)$ *we compute*

$$x = \tilde{s} - e' A^\top = (1,0,3,1) - (e_1, e_2) \begin{pmatrix} 0 & 0 & 1 & 2 & 3 & 1 \\ 0 & 0 & 2 & 4 & 1 & 2 \\ 0 & 0 & 3 & 3 & 2 & 4 \\ 0 & 0 & 1 & 2 & 1 & 3 \end{pmatrix}^\top$$

*and check if it has the remaining weight* $t - w = 1$.

*For* $e' = (e_1, e_2) = (1,0,0,3,0,0)$ *we get* $x = (0,3,4,0)$ *which is not of weight* $1$, *for* $e' = (3,0,0,0,1,0)$ *we get* $x = (3,4,1,0)$ *and we continue until the very last collision, where* $e' = (0,0,2,0,0,3)$ *and we get* $x = (1,0,0,0)$.

*Hence, we set* $e = (x, e') = (1,0,0,0,0,0,2,0,0,3)$ *which is of weight* $t = 3$ *and such that* $eH^\top = s$.

Note that Stern's algorithm is always at least as fast as Prange, as it recovers Prange by setting $w = \ell = 0$.

### 9.5.1 Asymptotic Cost

An important aspect of ISD algorithms (apart from the cost) is their asymptotic cost. The idea of the asymptotic cost is that we are interested in the exponent $e(R,q)$ such that for large $n$ the cost of the algorithm is given by $q^{(e(R,q)+o(1))n}$. This is crucial in order to compare different algorithms.

We consider codes of large length $n$, and consider the dimension and the error correction capacity as functions in $n$, for which we define

$$\begin{aligned}
\lim_{n\to\infty} t(n)/n &= T,\\
\lim_{n\to\infty} k(n)/n &= R.
\end{aligned}$$

Recall that $H_q(x) = x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x)$ and due to the GV bound we can set $T = \delta/2$, where $\delta = H_q^{-1}(1-R)$. If $c(n,k,t,q)$ denotes the cost of an algorithm, for example Prange's algorithm, then we are now interested in

$$e(R,q) = \lim_{n\to\infty} \frac{1}{n}\log_q(c(n,k,t,q)).$$

For this we often use Stirlings formula, that is if

$$\lim_{n\to\infty} a(n)/n = A,$$
$$\lim_{n\to\infty} b(n)/n = B$$

then

$$\lim_{n\to\infty} \frac{1}{n}\log_q\binom{a}{b} = A\log_q(A) - B\log_q(B) - (A-B)\log_q(A-B).$$

**Theorem 9.16.** *The asymptotic cost of Prange's algorithm is $q^{(e(q,R)+o(1))n}$, where*

$$e(q,R) = -(1-T)\log_q(1-T) - (1-R)\log_q(1-R) + (1-R-T)\log_q(1-R-T),$$

*where $T = H_q^{-1}(1-R)/2$.*

*Proof.* Recall that the cost of Prange's algorithm is given by

$$c(n,k,t,q) = \binom{n-k}{t}^{-1}\binom{n}{t}.$$

Using Stirling's formula, we get that

$$\begin{aligned}
&\lim_{n\to\infty} \frac{1}{n}\log_q\left(\binom{n-k}{t}^{-1}\binom{n}{t}\right) = \\
&-((1-R)\log_q(1-R) - T\log_q(T) - (1-R-T)\log_q(1-R-T)) \\
&+1\log_q(1) - T\log_q(T) - (1-T)\log_q(1-T) \\
&= -(1-T)\log_q(1-T) - (1-R)\log_q(1-R) + (1-R-T)\log_q(1-R-T).
\end{aligned}$$

Finally, due to the GV we have that $T = H_q^{-1}(1-R)/2$. □

**Exercise 9.17.** *Prove that the asymptotic cost of Prange is equal to*

$$H_q(T) - (1-R)H_q(T/(1-R)).$$

Let us also compute the asymptotic cost of Stern. Since we have internal parameters $\ell, w$ we first need to set

$$\lim_{n\to\infty} \ell(n)/n = L,$$
$$\lim_{n\to\infty} w(n)/n = W.$$

**Theorem 9.18.** *The asymptotic cost of Stern's algorithm is $q^{(e(q,R)+o(1))n}$, where*

$$e(q,R) = \min_{L,W}\left\{-2A - B + C + \max\left\{A + \frac{W}{2}\log_q(q-1), 2A + (W-L)\log_q(q-1)\right\}\right\},$$

*where*

$$\begin{aligned}
A =& \frac{R+L}{2}\log_q\left(\frac{R+L}{2}\right) - \frac{W}{2}\log_q\left(\frac{W}{2}\right) - \frac{R+L-W}{2}\log_q\left(\frac{R+L-W}{2}\right),\\
B =& (1-R-L)\log_q(1-R-L) - (T-W)\log_q(T-W)\\
&- (1-R-L-T+W)\log_q(1-R-L-T+W),\\
C =& -T\log_q(T) - (1-T)\log_q(1-T).
\end{aligned}$$

*Proof.* Recall that the cost of Stern's algorithm is given by

$$\begin{aligned}
c(n,k,t,q) =& \binom{(k+\ell)/2}{w/2}^{-2}\binom{n-k-\ell}{t-w}^{-1}\binom{n}{t}\\
&\cdot\left(\binom{(k+\ell)/2}{w/2}(q-1)^{w/2} + \binom{(k+\ell)/2}{w/2}^{2}(q-1)^{w-\ell}\right).
\end{aligned}$$

We start by computing

$$\begin{aligned}
A &= \lim_{n\to\infty}\frac{1}{n}\log_q\left(\binom{(k+\ell)/2}{w/2}\right)\\
&= \frac{R+L}{2}\log_q\left(\frac{R+L}{2}\right) - \frac{W}{2}\log_q\left(\frac{W}{2}\right) - \frac{R+L-W}{2}\log_q\left(\frac{R+L-W}{2}\right).
\end{aligned}$$

We then compute

$$\begin{aligned}
B &= \lim_{n\to\infty}\frac{1}{n}\log_q\left(\binom{n-k-\ell}{t-w}\right)\\
&= (1-R-L)\log_q(1-R-L) - (T-W)\log_q(T-W)\\
&\quad - (1-R-L-T+W)\log_q(1-R-L-T+W).
\end{aligned}$$

Finally,

$$C = \lim_{n\to\infty}\frac{1}{n}\log_q\left(\binom{n}{t}\right) = -T\log_q(T) - (1-T)\log_q(1-T).$$

Thus, we get that

$$\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n}\log_q\left(c(n,k,t,q)\right)\\
&= -2A - B + C + \max\left\{A + \frac{W}{2}\log_q(q-1), 2A + (W-L)\log_q(q-1)\right\}.
\end{aligned}$$

Since the algorithm will optimize the choices of $\ell, w$ with the restrictions

$$\ell < n - k - t + w, \qquad w < t.$$

□

We can then plot the cost for a fixed $q = 2$, as

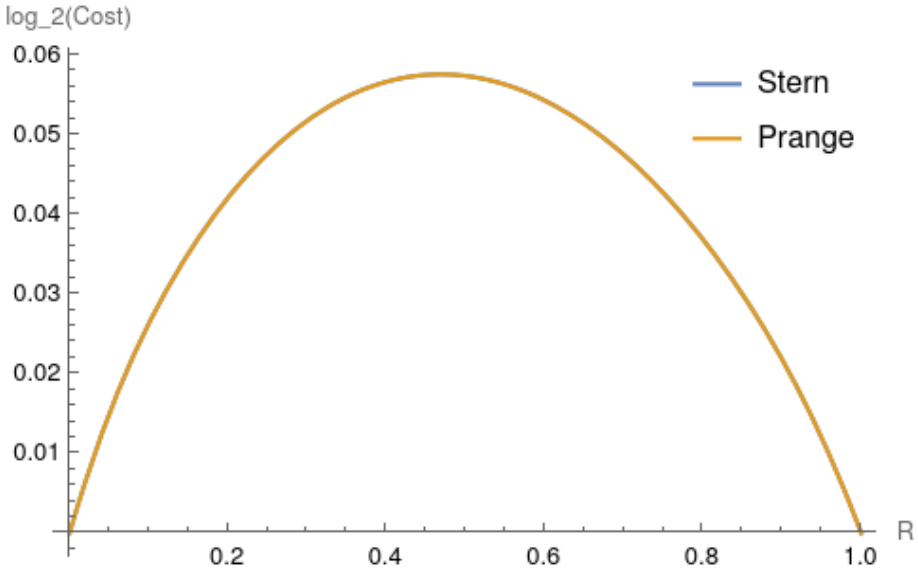



**Figure 1:** Comparison plot of Stern vs. Prange, $q = 2$

Their difference is barely visible. This changes if we also include newer ISD algorithms, such as MMT [11].

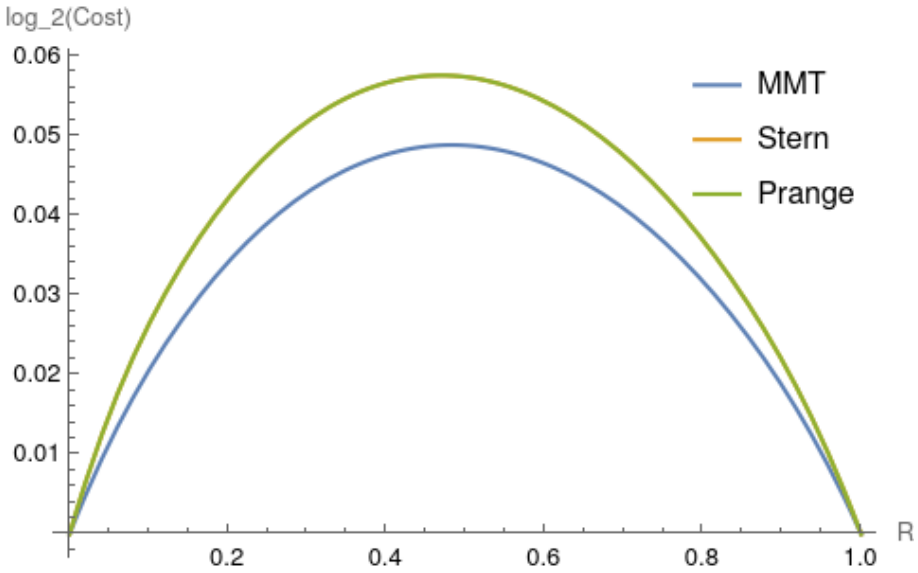



**Figure 2:** Comparison plot of MMT vs. Stern vs. Prange, $q = 2$

We often also give the maximal cost over all rates, that is

$$e^*(q) = \max\{e(R, q) \mid R \in [0, 1]\}.$$

We then get for $q = 2$ that

| Algorithm | $e^*(q)$ |
|---|---|
| Prange | 0.05747 |
| Stern | 0.05563 |

**Table 3:** Comparison of Prange and Stern for $q = 2$.

Clearly, there are more improvements to the simple ideas of Prange and Stern, but most rely on the explained steps and have a similar cost analysis. In fact, over the last 60 years, the exponent $e^*(q)$ has only decreased from Prange's $0.05747$ to $0.0473$ [12].

Although this research area is active and important for code-based cryptography there are many unsolved questions:

- How to decode a (quasi-)cyclic code?
- How to decode a $q$-ary code (faster)?
- How to decode for large weights?

Note that the code-based cryptosystem to be standardized (by the U.S. authorities) is HQC [**?**], which relies on quasi-cyclic codes. None of the ISD algorithms (until now) is able to incorporate this additional structure to lower its cost.

New proposals use codes over $\mathbb{F}_{2^m}$ and while the algorithms we have seen so far are also able to decode such $q$-ary codes, none of them use the additional structure of the extension field $\mathbb{F}_{2^m}$.

Additionally, for large $q$, the best decoder is the simple algorithm by Prange. In fact, all other decoders involve an enumeration step (that is build a list of vectors in $\mathbb{F}_q^{n'}$ of some weight $w$). As such lists have size $\binom{n'}{w}(q-1)^w$, the cost of such algorithms quickly grows too large. On the other hand, Prange's algorithm is oblivious of the underlying field.

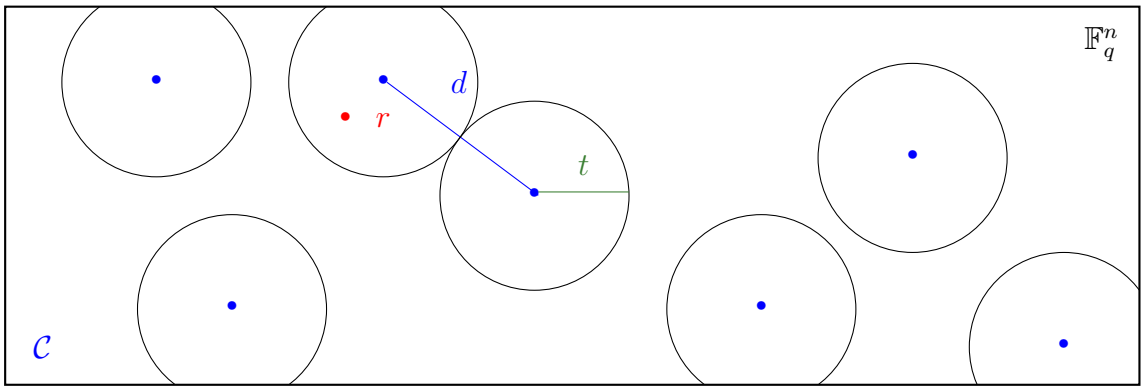
- The *Decoding Problem (DP)* is to decode a random linear code and is NP-hard.
- The DP is equivalent to the *Syndrome Decoding Problem (SDP)* and to the *Codeword Finding Problem (CFP)*.
- *Information Set Decoding (ISD)* algorithms decode a random linear code.
- ISD algorithms use information sets and assume a weight distribution of the error vector.
- ISD algorithms have exponential cost.

# 10 List Decoding

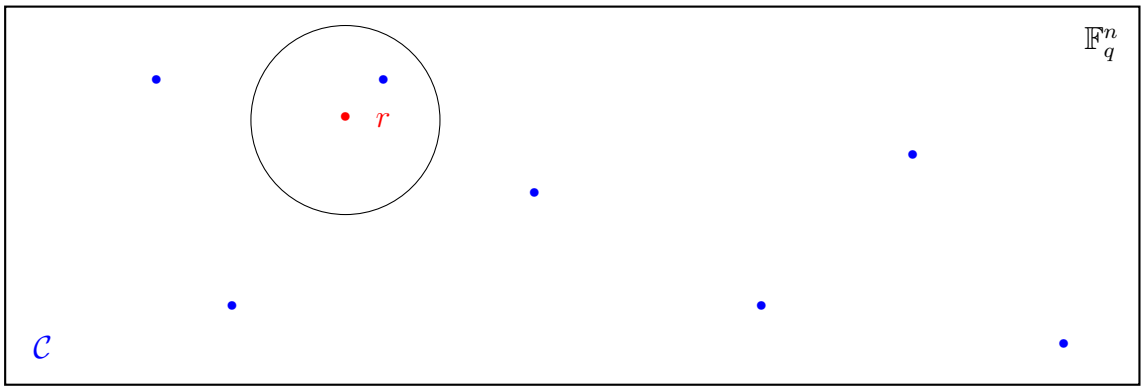
Let us revisit our decoding problem: previously we were given $r \in \mathbb{F}_q^n$ and wanted to find *the unique* codeword $c \in \mathcal{C}$ which is such that $d_H(r, c) \leq t$ and to do this we needed $t \leq \lfloor \frac{d-1}{2} \rfloor$.

In this chapter we will relax the condition of having a unique closest codeword.

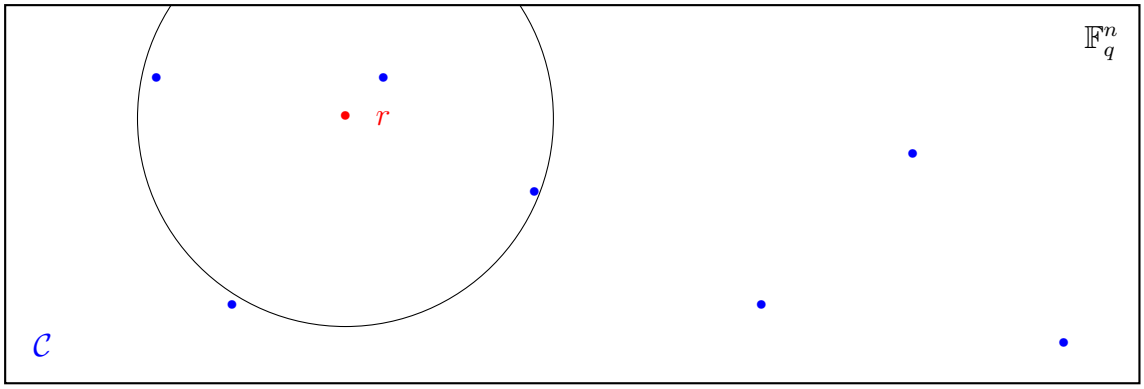
If we go back to our geometric interpretation, we had before, that is:



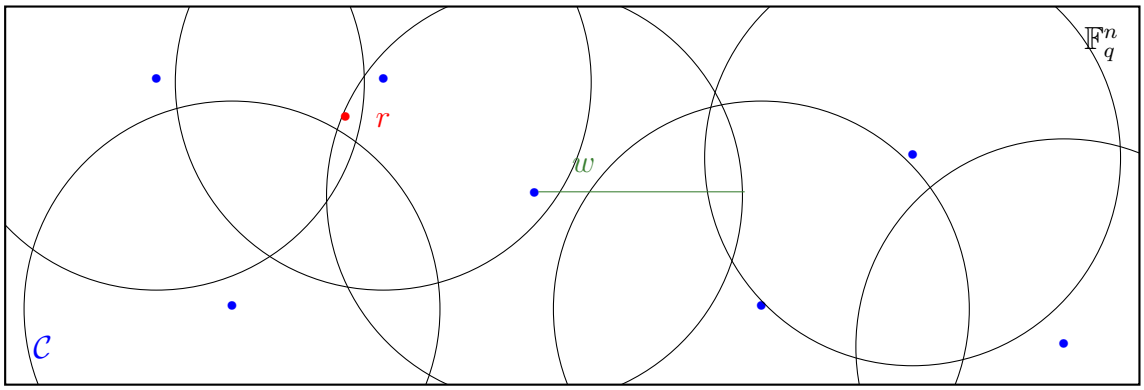

In this scenario, we wanted that the balls of radius $t$ around codewords of $\mathcal{C}$ do not intersect, so that $r$ lands in exactly one ball. Equivalently, there is only one codeword in the ball of radius $t$ around $r$ :




However, we can allow for a larger distance, say $w > t$, paying the price in having several codewords $c_i$ in the Hamming ball of radius $w$ around $r$.




Equivalently, the balls of radius $w$ around the codewords will intersect; thus $r$ lives in the ball of several codewords:

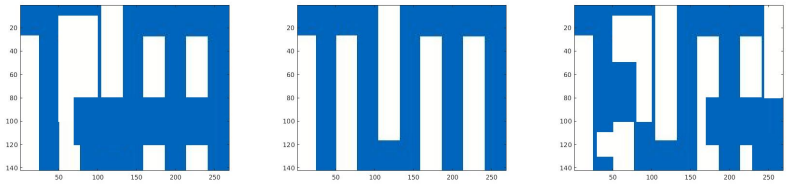



Assume that we have access to a decoder outputting all codewords in the ball of radius $w$ around $r$

$$\mathcal{L} = \{c \in \mathcal{C} \mid d_H(r, c) \leq w\} = \mathcal{C} \cap B_H(w, n, q, r).$$

We want that this list of this possibly sent codewords is not too large. For example, exponentially large lists would not be feasible, since also a decoder, having to output all these codewords immediately becomes infeasible, but a polynomial-sized list we can handle.

In practice, a receiver also has some "soft information" on what the sent message could be. For example, if our list of possible sent codewords encodes the information of



we might easily find the right message.

This motivates the following definition

**Definition 10.1.** Let $\mathcal{C}$ be a $[n,k]_q$ linear code. We say that $\mathcal{C}$ is $(w,L)$- *list decodable*, if for all $r \in \mathbb{F}_q^n$ we have that

$$L = |\{c \in \mathcal{C} \mid d_H(r,c) \leq w\}|.$$

Thus, this is a generalization of our previous scenario, where we considered $L = 1$, and $w = t = \lfloor \frac{d-1}{2} \rfloor$.

The first question we have to ask is: until which radius $w$ can we go, such that $L$ is still polynomial in $n$?

## 10.1 Johnson Bound

This question is answered by the Johnson bound.

**Theorem 10.2.** *Let $q$ be a prime power, $k, d \leq n$ and $\mathcal{C}$ be $[n,k,d]_q$ linear code. Define*

$$J_q(n,d) = n\frac{q-1}{q}\left(1 - \sqrt{1 - \frac{q}{q-1}\frac{d}{n}}\right).$$

*Then if $w < J_q(n,d)$ we have that $\mathcal{C}$ is $(w, qnd)$ -list decodable.*

For $q = 2$, we get

$$J_2(n,d) = \frac{n}{2}\left(1 - \sqrt{1 - 2\frac{d}{n}}\right).$$

*Proof.* We will only prove the statement for $q = 2$, as it is already complicated enough for this case.

Given $r \in \mathbb{F}_2^n$ and denote by $\mathcal{L}$ the codewords at radius $\leq w$ from $r$ that is

$$\mathcal{L} = \mathcal{C} \cap B_H(w,n,q,r),$$

and denote its size by $L$.

We now construct a bipartite graph $\mathcal{G} = (P_1 \cup P_2, E)$, where we can partition the nodes into two sets $P_1, P_2$ and only have edges between nodes from $P_1$ and $P_2$, but for example no edges between nodes from $P_1$ and $P_1$. We set the following rules:

- Each $c \in \mathcal{L}$ is a node in $P_1$ and each entry $r_i$ is a node in $P_2$.
- We put an edge between $c$ and $r_j$ if and only if $c_j = r_j$.

Let us denote by $d_j$ the *degree* of the node $r_j$, that is: in how many edges is $r_j$.

We say we have an *angle* in the graph, when we have two codewords $c, c'$ with $(c, r_j) \in E, (c', r_j) \in E$.

Since the codewords $c \in \mathcal{L}$ all have distance $\leq w$ from $r$, we know that for each $c \in \mathcal{L}$, it must have $\geq n - w$ edges to $r_1, \ldots, r_n$. Thus, if we sum all edges, we get

$$\sum_{j=1}^{n} d_j \geq (n-w)L.$$

Let us count the number of angles in the graph. Since each $r_j$ has $d_j$ edges going out of $r_j$, we have to select two edges, giving

$$\sum_{j=1}^{n} \binom{d_j}{2}.$$

$$\sum_{j=1}^{n} \binom{d_j}{2} = \sum_{j=1}^{n} \frac{(d_j-1)d_j}{2} = \frac{1}{2}\left(\sum_{j=1}^{n} d_j^2 - \sum_{j=1}^{n} d_j\right).$$

Due to the Cauchy-Schwarz inequality, we have that

$$\sum_{j=1}^{n} d_j^2 \geq \frac{1}{n}\left(\sum_{j=1}^{n} d_j\right)^2$$

and hence

$$\begin{aligned}\sum_{j=1}^{n} \binom{d_j}{2} &\geq \frac{1}{2}\left(\frac{1}{n}\left(\sum_{j=1}^{n} d_j\right)^2 - \sum_{j=1}^{n} d_j\right) \\ &= \frac{1}{2}\left(\sum_{j=1}^{n} d_j\right)\left(\frac{1}{n}\sum_{j=1}^{n} d_j - 1\right).\end{aligned}$$

Since $\sum_{j=1}^{n} d_j \geq (n-w)L$, we get that

$$\sum_{j=1}^{n} \binom{d_j}{2} \geq \frac{1}{2}\left(\frac{1}{n}(n-w)^2L^2 - (n-w)L\right).$$

We can also count the number of angles using $c$ : Fixed $c, c' \in \mathcal{L}$ can only agree in at most $n-d$ entries, since $d_H(c, c') \geq d$. Hence, the number of angles between the same $c, c'$ cannot be larger than $n-d$, that is for each pair $(c, c')$ we have at most $n-d$ angles:

Hence we have

$$\sum_{j=1}^{n} \binom{d_j}{2} \leq \binom{L}{2}(n-d) = \frac{1}{2}(n-d)(L^2-L).$$

Hence,

$$\frac{1}{2}\left(\frac{1}{n}(n-w)^2L^2-(n-w)L\right)\leq\frac{1}{2}(n-d)(L^2-L).$$

Which implies that

$$L^2\left(\frac{1}{n}(n-w)^2-(n-d)\right)\leq L(n-w-n+d)=L(d-w),$$

and thus if $\frac{1}{n}(n-w)^2\neq(n-d)$, we get

$$L\leq\frac{d-w}{\frac{1}{n}(n-w)^2-(n-d)}=\frac{\frac{d}{n}-\frac{w}{n}}{(1-\frac{w}{n})^2-(1-\frac{d}{n})}.$$

Thus, if $(1-\frac{w}{n})^2>(1-\frac{d}{n})$, then $L$ must be bounded, in particular $L\leq(d-w)/n$.

This is equivalent to $w<n\left(1-\sqrt{1-\frac{d}{n}}\right)\leq J_2(n,d)$. □

We note that within this proof we have discovered a universal bound oblivious of $q$, that is

$$J(n,d)=n\left(1-\sqrt{1-\frac{d}{n}}\right).$$

More precisely, for all $0\leq x\leq 1-1/q$ it holds that

$$1-\sqrt{1-x}\leq(1-1/q)\sqrt{1-\frac{q}{q-1}x}.$$

Hence if we use this in out $J_q(n,d)$, by setting $x=d/n$ we get that

$$J(n,d)=n\left(1-\sqrt{1-\frac{d}{n}}\right)\leq(1-1/q)\sqrt{1-\frac{q}{q-1}\delta}=J_q(n,d).$$

**Definition 10.3.** Let $d\leq n$ be positive integers. The *Johnson radius* is given by

$$J(n,d)=n\left(1-\sqrt{1-\frac{d}{n}}\right)=n-\sqrt{n(n-d)}.$$

**Corollary 10.4.** *Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code and let $w<J(n,d)$. Then $\mathcal{C}$ is $(w,qdn)-$list decodable.*

Equivalently, any ball of radius up to $J(n,d)$ contains at most poly$(qn)$ codewords.
For Reed-Solomon codes, this means that

$$J(n,n-k+1)=n-\sqrt{n(k-1)}\sim n(1-\sqrt{R}).$$

This is much larger than the unique decoding radius $t=\frac{n-k}{2}=n(1-R)/2$.

## 10.2 Bivariate Polynomials

Before we can start with a list decoder for Reed-Solomon codes, let us recall some useful facts of bivariate polynomials.

**Definition 10.5.** A *bivariate polynomial* $Q(x,y) \in \mathbb{F}_q[x,y]$ is a polynomial in two variables $x, y$ and given by

$$Q(x,y) = \sum_{i=0}^{d_x} \sum_{j=0}^{d_y} q_{i,j} x^i y^j,$$

with $q_{i,j} \in \mathbb{F}_q$. We say that $Q(x,y)$ has $x$-degree $\deg_x(Q(x,y)) = d_x$ and $y$-degree $\deg_y(G(x,y)) = d_y$. For a monomial $x^i y^j$, we say it has degree $i+j$. The *total* degree of $(Q(x,y))$ is given by the largest $i+j$, for which $q_{i,j} \neq 0$.

We can write a bivariate polynomial also as univariate polynomial in the polynomial ring, that is $Q(x,y) \in (\mathbb{F}_q[x])[y]$, by gathering all the monomials of the same $y$-degree, i.e.,

$$Q(x,y) = \sum_{j=0}^{d_y} Q_j(x) y^j,$$

where $Q_j(x) \in \mathbb{F}_q[x]$.

**Example 10.6.** *Let us consider* $Q(x,y) = (x^2 - y)(y - y) = x^3 - xy - x^2 y + y^2$. *Then*

$$Q(x,y) = x^3 \cdot y^0 + (-x - x^2) y^1 + 1 \cdot y^2,$$

*and hence we can set*

$$Q_0(x) = x^3, Q_1(x) = -x - x^2, Q_3(x) = 1.$$

As for univariate polynomials, we have that their roots correspond to factors:

**Proposition 10.7.** *Let* $Q(x,y) \in \mathbb{F}_q[x,y]$ *and let* $f(x) \in \mathbb{F}_q[x]$. *Then* $Q(x, f(x)) = 0$ *if and only if* $(y - f(x)) \mid Q(x,y)$.

**Example 10.8.** *In our previous example, that is* $Q(x,y) = x^3 - xy - x^2 y + y^2$, *we plug in* $y = x^2$ *and get that +*

$$Q(x, x^2) = x^3 - x^3 - x^4 + x^4 = 0.$$

*Hence* $(y - x^2) \mid Q(x,y)$, *which is true as* $Q(x,y) = (x^2 - y)(y - y) = x^3 - xy - x^2 y + y^2$.

Similar to univariate polynomials, we have that bivariate polynomials can not have more roots than their total degree.

**Proposition 10.9.** *Let* $Q(x,y) \in \mathbb{F}_q[x,y]$ *with* $\deg_y(Q(x,y)) = d$. *Then there exist at most* $d$ *many polynomials* $f(x) \in \mathbb{F}_q[x]$ *with* $Q(x, f(x)) = 0$.

*Proof.* If $Q(x, f_i(x)) = 0$ for $i \in \{1, \ldots, d+1\}$, then by Proposition 10.7, we have that

$$\prod_{i=1}^{d+1} (y - f_i(x)) \mid Q(x,y).$$

However, as the left hand has $y$-degree $d+1$, while the right hand has $y$-degree $d$, we get a contradiction. □

## 10.3 Recap on Berlekamp-Welch

In this section, we want to list-decode RS codes. Thus, we let $\alpha \in \mathbb{F}_q^n$ consist of all distinct entries and define $\mathcal{C} = \mathcal{RS}_{q,n,k}(\alpha)$.

If we receive $r \in \mathbb{F}_q^n$, the task is to efficiently recover *all* $c \in \mathcal{C}$ with $d_H(r, c) \leq w$, for some $w \leq n(1-\sqrt{R})$, as per the Johnson bound. This is equivalent to finding a polynomial $f(x) \in \mathbb{F}_q[x]$ of degree $\deg(f(x)) \leq k-1$, such that $f(\alpha_i) = r_i$ for at least $n - w$ many $i \in \{1, \ldots, n\}$.

In Chapter 3, we decoded uniquely up to $w = \lfloor \frac{d-1}{2} \rfloor$, using the Berlekamp-Welch decoder. And while there are more efficient decoders, this one is perfect for educational reasons: we can generalize this unique decoder to a list decoder.

Recall that in the Berlekamp-Welch algorithm, we recover $f(x)$ by performing two steps:

1. Find $E(x), N(x)$ (of degree $t$, respectively $k+t-1$) which are such that $E(\alpha_i)r_i = N(\alpha_i)$ for all $i \in \{1, \ldots, n\}$.
2. Compute $N(x)/E(x) = f(x)$.

This works as $N(x) = E(x)f(x)$ and $E(x)$ is the *error locator polynomial,* that is $E(\alpha_i) = 0$ for all $i \in \text{supp}_H(e)$, that is whenever $r_i \neq f(\alpha_i)$.

We can reformulate this algorithm using bivariate polynomials.

1. Find $Q(x, y)$ (of some appropriate degree which we will clarify later) such that $Q(\alpha_i, r_i) = 0$ for all $i \in \{1, \ldots, n\}$.
2. Find $f(x) \in \mathbb{F}_q[x]$ of degree $< k$ such that $Q(x, f(x)) = 0$.

By setting $Q(x, y) = E(x)y - N(x)$ we recover the original Berlekamp-Welch algorithm.

In fact, if we have found the correct $Q(x, y)$, that is $Q(x, y) = E(x)y - N(x)$, then $f(x) = N(x)/E(x)$ is such that $Q(x, f(x)) = 0$.

## 10.4 Sudan's Algorithm

By relaxing the form of $Q(x, y) = Q_0(x) + yQ_1(x)$ (where $Q_0(x) = N(x)$ and $Q_1(x) = E(x)$ before) and allowing also larger degrees of $y$, we are able to handle more than just $t = \lfloor \frac{d-1}{2} \rfloor$ errors.

The algorithm we present here is by Sudan [18], and is able to list-decode an error of weight up to

$$w = \lfloor n - 2\sqrt{nk} \rfloor.$$

Note that $w/n = 1 - 2\sqrt{R} < 1 - \sqrt{R}$ and the algorithm does not reach yet the possible Johnson radius. This was later solved by the Guruswami-Sudan algorithm [6]. For this lecture, we will already be content with Sudan's list decoder.

**Algorithm 5** Sudan's Algorithm

Input: the evaluation points $\alpha \in \mathbb{F}_q^n$, the received vector $r \in \mathbb{F}_q^n$, $\ell = \sqrt{nk}$.
Output: $f(x) \in \mathbb{F}_q[x]$, such that $d_H(f(\alpha), r) \leq n - 2\ell$.

1: Find $Q(x, y) \in \mathbb{F}_q[x, y]$ of $\deg_x(Q(x, y)) \leq \ell, \deg_y(Q(x, y)) \leq n/\ell$ such that

$$Q(\alpha_i, r_i) = 0 \; \forall \; i\{1, \ldots, n\}.$$

2: Find all $f_i(x) \in \mathbb{F}_q[x]$ of degree $\deg(f_i) \leq k - 1$ such that

$$Q(x, f_i(x)) = 0.$$

3: return all $c_i = (\beta_1 f_i(\alpha_1), \ldots, \beta_n f_i(\alpha_n))$.

In the algorithm we set the $x$-degree of $Q(x, y)$ to be $\ell = \sqrt{nk}$ and will later see why this makes sense. The algorithm of Sudan then works as follows.

We see that we have not changed the second step, or the condition of the first step, i.e., $Q(\alpha_i, r_i) = 0$ for all $i\{1, \ldots, n\}$. The only difference to the Berlekamp-Welch decoder lies in the degree of $Q(x, y)$.

How do we perform these two steps?

The first step can be solved as before, by setting up a linear system of equations and solving it.

In order for such a system to have a non-trivial solution we require that the number of variables is greater than or equal to the number of equations. The number of equations is still $n$, as $Q(\alpha_i, r_i) = 0$ should hold for $i \in \{1, \ldots, n\}$. The number of variables is the number of coefficients $q_{i,j}$ in the bivariate polynomial $Q(x, y)$. As $Q(x, y)$ has $x$-degree $\leq \ell$ and $y$-degree $\leq n/\ell$, we get that the number of coefficients is $\leq (\ell + 1)(n/\ell + 1) = n + \ell + n/\ell + 1$.

Thus, we get that the total degree of $Q(x, y)$ is at most $n + \sqrt{nk} + \sqrt{R}^{-1} + 1$, which is indeed always greater than $n$.

The second step still makes sense, that is if $f(x)$ is the polynomial belonging to a possible codeword, that is $\deg(f) < k$ and $f(\alpha_i) = r_i$ for more than $2\ell$ many $i$, then

$$Q(x, f(x)) = 0.$$

**Proposition 10.10.** *Let $f(X) \in \mathbb{F}_q[x]$ be such that $\deg(f) < k$ and $f(\alpha_i) = r_i$ for more than $2\ell$ many $i$, then*

$$Q(x, f(x)) = 0.$$

*Proof.* Let us define $R(x) = Q(x, f(x))$. In order to show that $R(x) = 0$, we want to show that $R(x)$ has more roots than its degree allows.

Thus, we start by computing the degree as

$$\deg(r) \leq \deg_x(Q) + \deg(f)\deg_y(Q) < \ell + kn/\ell = 2\sqrt{nk},$$

which follows again by the choice of $\ell = \sqrt{nk}$. Here we also see the reasoning for setting $\ell = \sqrt{nk}$: this choice balances the two degrees out.

Next, we check in how many $\alpha_i$ the polynomial $R$ vanishes. Since $f(\alpha_i) = r_i$ for more than $2\sqrt{nk}$ many $i$, in those positions $i$ we get that

$$R(\alpha_i) = Q(\alpha_i, f(\alpha_i)) = Q(\alpha_i, r_i) = 0.$$

Hence $\deg(R) < 2\sqrt{nk}$, while $R(x)$ has more than $2\sqrt{nk}$ roots.

□

There are at most $\deg_y(Q) = n/\ell = \sqrt{n/k}$ many polynomials $f(x)$ which are such that $Q(x, f(x)) = 0$ and thus our list has size at most

$$L \leq \sqrt{1/R},$$

and hence if $R$ is a constant, then our list size $L$ is also a constant.

Unfortunately, there is no short example for this list decoder. In fact, in order to have that the list decoding radius of Sudan's algorithm is greater than the unique decoding radius, we want that

$$w = \lfloor n - 2\sqrt{nk} \rfloor > t = \lfloor \frac{n-k}{2} \rfloor,$$

which is equivalent to

$$n(1 - 2\sqrt{R}) > n(1-R)/2,$$

which only holds for very small rates, i.e., $R < 0.07$ and thus to have at least $k = 2$, we already need $n = 29$.

- A code is called $(w, L)$-*list decodable* if $|B_H(w, n, q) \cap \mathcal{C}| = L$.
- The *Johnson bound* tells us that for any $[n, k, d]_q$ linear code if $w < J_q(n, d)$, then $\mathcal{C}$ is $(w, qnd)$-list decodable.
- The *Johnson radius* $J(n, d) = n - \sqrt{n(n-d)}$ is such that any $[n, k, d]_q$ linear code is $(w, qnd)$-list decodable for $w < J(n, d)$.
- Sudan's list-decoding algorithm is a generalization of the Berlekamp-Welch algorithm.
- Sudan's algorithm can correct up to $w = \lfloor n - 2\sqrt{nk} \rfloor$ errors and produces a list of size $L \leq \sqrt{1/R}$.

# 11 MacWilliams Identity

In this chapter we show one of the most elegant results in coding theory: the MacWilliams identities.

For this recall the definition of the weight enumerator: Let $\mathcal{C}$ be a $[n,k,d]_q$ linear code. Then for a weight $0 \leq w \leq n$ the weight enumerator of $\mathcal{C}$ is given by

$$A_w(\mathcal{C}) = |\{c \in \mathcal{C} \mid \mathrm{wt}_H(c) = w\}|.$$

Clearly, we know some of the weight enumerators: $A_0(\mathcal{C}) = 1$ and since $d_H(\mathcal{C}) = d$ we also have $A_i(\mathcal{C}) = 0$ for all $0 < i < d$.

If we are interested in all weights $w$, that is $(0, 1, \ldots, n)$ we call this the *weight distribution*.

**Definition 11.1.** Let $\mathcal{C}$ be a $[n,k]_{p^m}$ linear code. The *weight distribution* of $\mathcal{C}$ is given by

$$(A_0(\mathcal{C}), A_1(\mathcal{C}), \ldots, A_n(\mathcal{C})).$$

Although the weight enumerator of a code is a priori independent of the weight enumerator of its dual, their weight distributions are not.

**Theorem 11.2** (MacWilliams Identities)**.** *Let $\mathcal{C}$ be a $[n,k]_{p^m}$ linear code. Let $0 \leq w \leq n$. Then*

$$A_w(\mathcal{C}^\perp) = \frac{1}{|\mathcal{C}|} \sum_{v=0}^{n} A_v(\mathcal{C}) \sum_{s=0}^{w} \binom{n-v}{w-s} \binom{v}{s} (-1)^s (p^m - 1)^{w-s}.$$

Hence if we are given the weight distribution of $\mathcal{C}$ we also know the weight distribution of $\mathcal{C}^\perp$.

In order to prove this beautiful result, we first need a few ingredients, namely: characters, the Schur orthogonality and the Krawtchouk coefficients.

## 11.1 Characters

In all generality, a group character defines a representation of a group $G$ in terms of a complex function.

**Definition 11.3.** Let $(G, +)$ be a group and denote by $\mathbb{C}^\star = \{z \in \mathbb{C} \mid |z| = 1\}$. A *character* is a function

$$\chi : G \to \mathbb{C}^\star,$$

such that

$$\chi(a + b) = \chi(a)\chi(b)$$

for all $a, b \in G$.

We further say that $\chi$ is *principal* is $\chi(x) = 1$ for all $x \in G$. Clearly, we are usually interested in non-principal characters.

**Exercise 11.4.** *Let $(G,+)$ be a group and $\chi : G \to \mathbb{C}^\star$ a character. Show that $\chi(0) = 1$.*

There is a large theory on characters, but in the case where $(G,+) = (\mathbb{F}_{p^m}^n, +)$ we know exactly how they look like.

**Proposition 11.5.** *Let $\zeta$ be a $p$th root of unity. Then any character $\chi : \mathbb{F}_{p^m}^n \to \mathbb{C}^\star$ is given by*

$$\chi_a(x) = \zeta^{Tr(\langle x,a\rangle)},$$

*for some $a \in \mathbb{F}_{p^m}^n$.*

We note that if $a = 0$, then $\chi_a$ is a principal character.

## 11.2 Schur Orthogonality

The Schur orthogonality is an important property of characters. We start with the easy case of $G = \mathbb{F}_{p^m}$.

**Lemma 11.6.** *Let $\chi$ be a character of $\mathbb{F}_{p^m}$, then*

$$\sum_{x\in\mathbb{F}_{p^m}} \chi(x) = \begin{cases} p^m & \text{if } \chi \text{ is principal,} \\ 0 & \text{else.} \end{cases}$$

*Proof.* If $\chi$ is principal then

$$\sum_{x\in\mathbb{F}_{p^m}} \chi(x) = \sum_{x\in\mathbb{F}_{p^m}} 1 = p^m.$$

If $\chi$ is non-principal, that is there exists some $b \in \mathbb{F}_{p^m}$ such that $\chi(b) \neq 1$, then

$$\sum_{x\in\mathbb{F}_{p^m}} \chi(x) = \sum_{x\in\mathbb{F}_{p^m}} \chi(x+b) = \sum_{x\in\mathbb{F}_{p^m}} \chi(x)\chi(b) = \chi(b) \sum_{x\in\mathbb{F}_{p^m}} \chi(x).$$

Since $\chi(b) \neq 1$, this implies that $\sum_{x\in\mathbb{F}_{p^m}} \chi(x) = 0$. □

**Corollary 11.7.** *Let $\chi$ be a non-principal character. Then*

$$\sum_{x\in\mathbb{F}_{p^m}^\star} \chi(x) = -1.$$

We can extend this result to codes.

**Lemma 11.8.** *Let $\mathcal{C}$ be a $[n,k]_{p^m}$ linear code and $\chi_a$ be a non-principal character. Then*

$$\sum_{c\in\mathcal{C}} \chi_a(c) = \begin{cases} |\mathcal{C}| & \text{if } a \in \mathcal{C}^\perp, \\ 0 & \text{else.} \end{cases}.$$

*Proof.* If $a \in \mathcal{C}^\perp$, then $\langle a, c\rangle = 0$ for all $c \in \mathcal{C}$, thus

$$\sum_{c\in\mathcal{C}} \chi_a(c) = \sum_{c\in\mathcal{C}} \zeta^{\mathrm{Tr}(\langle a,c\rangle)} = \sum_{c\in\mathcal{C}} \zeta^0 = \sum_{c\in\mathcal{C}} 1 = |\mathcal{C}|.$$

If $a \notin \mathcal{C}^\perp$, then there exists some $b \in \mathcal{C}$ such that $\langle a, b\rangle \neq 0$ and hence $\chi_a(b) = \zeta^{\mathrm{Tr}(\langle a,b\rangle)} \neq 1$, then

$$\sum_{c\in\mathcal{C}} \chi_a(c) = \sum_{c\in\mathcal{C}} \chi_a(c+b) = \sum_{c\in\mathcal{C}} \chi_a(c)\chi_a(b) = \chi_a(b)\sum_{c\in\mathcal{C}} \chi_a(c).$$

Since $\chi_a(b) \neq 1$, this implies that $\sum_{c\in\mathcal{C}} \chi_a(c) = 0$. □

For some set $A$ let us denote by $!_A(x)$ the indicator function, that is

$$\mathbb{1}_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{else.} \end{cases}.$$

We will rewrite this result as follows:

**Corollary 11.9.** *Let $\mathcal{C}$ be a $[n,k]_{p^m}$ linear code. Let $\chi$ be a non-principal character. Then*

$$\mathbb{1}_{\mathcal{C}^\perp}(x) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \chi_x(c).$$

## 11.3 Krawtchouk Coefficients

The way we want to prove the MacWilliams identities (spoilers ahead) is:

$$A_w(\mathcal{C}^\perp) = \sum_{x\in\mathbb{F}_{p^m}:\mathrm{wt}_H(x)=w} \mathbb{1}_{\mathcal{C}^\perp}(x) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \sum_{x\in\mathbb{F}_{p^m}:\mathrm{wt}_H(x)=w} \chi_x(c),$$

where we have used the Schur orthogonality.

Thus if, for all $c$ of weight $v$

$$\sum_{x\in\mathbb{F}_{p^m}:\mathrm{wt}_H(x)=w} \chi_x(c) = K_w(v)$$

then we get that

$$A_w(\mathcal{C}^\perp) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\sum_{v=0}^{n} \mathbb{1}_{\mathrm{wt}_H(c)=v} K_w(v) = \frac{1}{|\mathcal{C}|}\sum_{v=0}^{n} A_v(\mathcal{C})K_w(v).$$

Thus, we are left with showing that $K_w(v)$ exists, that is for all $c \in \mathbb{F}_{p^m}$ of weight $\mathrm{wt}_H(c) = v$ the quantity $\sum_{x\in\mathbb{F}_{p^m}^n:\mathrm{wt}_H(x)=w} \chi_x(c)$ is independent on the choice of $c$ and only depends on the weight $w$.

**Definition 11.10.** Let $w, v \in \{0, \dots, n\}$ and $c \in \mathbb{F}_{p^m}^n$ have weight $v$. The *Krawtchouk coefficient* is defined as

$$K_w(v) = \sum_{x \in \mathbb{F}_{p^m} : \mathrm{wt}_H(x) = w} \chi_x(c).$$

**Lemma 11.11.** *Let $w, v \in \{0, \dots, n\}$ and $c \in \mathbb{F}_{p^m}^n$ have weight $v$. The Krawtchouk coefficient is given by*

$$K_w(v) = \sum_{s=0}^{w} \binom{n-v}{w-s} \binom{v}{s} (-1)^s (p^m - 1)^{w-s}.$$

*Proof.* Let us fix $c \in \mathbb{F}_{p^m}^n$ of weight $v$. We will later see that the resulting formula is independent on the choice of $c$.

To compute

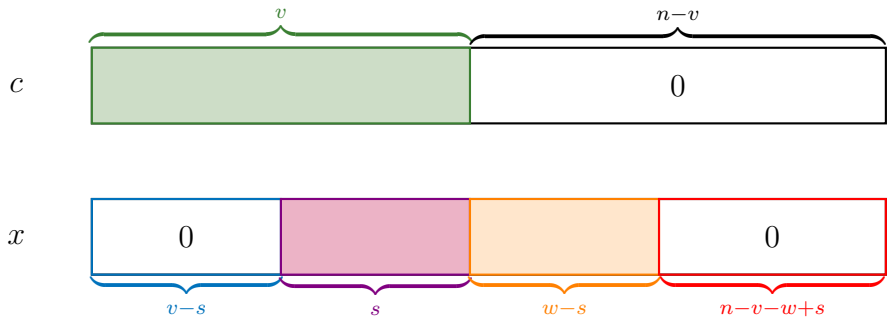
$$K_w(v) = \sum_{x \in \mathbb{F}_{p^m}^n : \mathrm{wt}_H(x) = w} \chi_x(c) = \sum_{x \in \mathbb{F}_{p^m}^n : \mathrm{wt}_H(x) = w} \zeta^{\mathrm{Tr}(\langle x, c \rangle)}$$

we observe that $\langle x, c \rangle = x_1 c_1 + \cdots + x_n c_n$ and thus $\mathrm{Tr}(\langle x, y \rangle) = \mathrm{Tr}(x_1 c_1) + \cdots + \mathrm{Tr}(x_n c_n)$ is defined componentwise.

Thus,

$$K_w(v) = \prod_{j=1}^{n} \sum_{x_j \in \mathbb{F}_{p^m}} \zeta^{\mathrm{Tr}(x_j c_j)}.$$

Since we have to go through all possible $x \in \mathbb{F}_{p^m}^n$ with $\mathrm{wt}_H(x) = w$, we have to go through all possible support overlaps of size $s \in \{0, \dots, w\}$:




This already implies we have $\binom{v}{s}\binom{n-v}{w-s}$ choices to place the support overlap.

Now in the first (blue) region of size $v - s$, we get in each entry

$$\sum_{x_j \in \mathbb{F}_{p^m}} \zeta^{\mathrm{Tr}(x_j c_j)} = \sum_{x_j = 0} \zeta^0 = 1.$$

In the second (purple) region of size $s$, we get in each entry

$$\sum_{x_j \in \mathbb{F}_{p^m}} \zeta^{\mathrm{Tr}(x_j c_j)} = \sum_{a \in \mathbb{F}_{p^m}^\star} \zeta^a = -1.$$

In the third (orange) region of size $w-s$, we get in each entry

$$\sum_{x_j \in \mathbb{F}_{p^m}} \zeta^{\mathrm{Tr}(x_j c_j)} = \sum_{x_j \in \mathbb{F}_{p^m}^\star} \zeta^0 = (p^m - 1).$$

Finally, in the fourth (red) region of size $n-v-w+s$, we get in each entry

$$\sum_{x_j \in \mathbb{F}_{p^m}} \zeta^{\mathrm{Tr}(x_j c_j)} = \sum_{x_j = 0} \zeta^0 = 1.$$

Hence if the support overlap size $s$ is fixed, then

$$\prod_{j=1}^{n} \sum_{x_j \in \mathbb{F}_q} \zeta^{\mathrm{Tr}(x_j c_j)} = (-1)^s (p^m - 1)^{w-s}.$$

Thus,

$$K_w(v) = \sum_{s=0}^{w} \binom{n-v}{w-s} \binom{v}{s} (-1)^s (p^m - 1)^{w-s}.$$

□

With this we can finally prove the MacWilliams identities.

*Proof.* Let us write

$$\begin{aligned}
A_w(\mathcal{C}^\perp) &= \sum_{x \in \mathbb{F}_{p^m} : \mathrm{wt}_H(x) = w} \mathbb{1}_{\mathcal{C}^\perp}(x) \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \sum_{x \in \mathbb{F}_{p^m} : \mathrm{wt}_H(x) = w} \chi_x(c). \\
&= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \sum_{v=0}^{n} \mathbb{1}_{\mathrm{wt}_H(c) = v} K_w(v) \\
&= \frac{1}{|\mathcal{C}|} \sum_{v=0}^{n} A_v(\mathcal{C}) K_w(v) \\
&= \frac{1}{|\mathcal{C}|} \sum_{v=0}^{n} A_v(\mathcal{C}) \sum_{s=0}^{w} \binom{n-v}{w-s} \binom{v}{s} (-1)^s (p^m - 1)^{w-s},
\end{aligned}$$

where we have used the Schur orthogonality and the Krawtchouk coefficients. □

Note that the MacWilliams identities are often formulated in its polynomial form, that is: we define the *weight enumerator polynomial* as

$$W_{\mathcal{C}}(x, y) = \sum_{i=}^{n} A_i x^{n-i} y^i,$$

then

$$W_{\mathcal{C}^\perp}(x,y) = \frac{1}{|\mathcal{C}|} W_{\mathcal{C}}(x + (p^m - 1)y, x - y).$$

**Example 11.12.** *Let us consider $\mathcal{C} = \langle (0,1,2) \rangle \subset \mathbb{F}_3^3$. Thus $A_0 = 1, A_1 = 0, A_2 = 2, A_3 = 0$. This is enough information to get the weight distribution of the dual.*

$$\begin{aligned}
A_0(\mathcal{C}^\perp) &= \frac{1}{3}\left(A_0\binom{0}{0}\binom{3}{0}2^0 + A_1 + A_2\binom{2}{0}\binom{1}{0}2^0 + A_3\right) \\
&= \frac{1}{3}\left(1 \cdot 1 + 0 + 2 \cdot 1 + 0\right) = \frac{1}{3}3 = 1, \\
A_1(\mathcal{C}^\perp) &= \frac{1}{3}\left(A_0\left(\binom{0}{0}\binom{3}{1}2^1 + 0\right) + A_1 + A_2\left(\binom{2}{0}\binom{1}{1}2^1 + \binom{2}{1}\binom{1}{0}(-1)2^0\right) + A_3\right) \\
&= \frac{1}{3}\left(1 \cdot (6+0) + 0 + 2 \cdot (2-2) + 0\right) = \frac{1}{3}6 = 2, \\
A_2(\mathcal{C}^\perp) &= \frac{1}{3}\left(A_0\left(\binom{0}{0}\binom{3}{2}2^2 + 0 + 0\right) + A_1 + A_2\left(0 + \binom{2}{1}\binom{1}{1}(-1)2^1 + \binom{2}{2}\binom{1}{0}2^0\right) + A_3\right) \\
&= \frac{1}{3}\left(1 \cdot 12 + 0 + 2 \cdot (-3) + 0\right) = \frac{1}{3}6 = 2, \\
A_3(\mathcal{C}^\perp) &= \frac{1}{3}\left(A_0\left(\binom{0}{0}\binom{3}{3}2^3 + 0 + 0 + 0\right) + A_1 + A_2\left(0 + 0 + \binom{2}{2}\binom{1}{1}2^1 + 0\right) + A_3\right) \\
&= \frac{1}{3}\left(1 \cdot 8 + 0 + 2 \cdot 2 + 0\right) = \frac{1}{3}12 = 4.
\end{aligned}$$

*Which is correct as $\mathcal{C}^\perp = \langle (1,0,0), (0,1,1) \rangle$.*

## 11.4 Linear Programming Bound

We may use these identities to get a bound on the size of the code. In fact, if $\mathcal{C}$ has weight distribution $(A_0, \ldots, A_n)$, then

$$|\mathcal{C}| = \sum_{i=}^{n} A_i.$$

This gives one of the tighest upper bounds on the size of a code with given length $n$ and minimum distance $d$.

**Theorem 11.13** (Linear Programming Bound)**.** *Maximize $\sum_{i=0}^{n} A_i$ under the linear constraint*

- $A_0 = 1$,
- $A_i \geq 0$ *for all* $i \in \{0, \ldots, n\}$,
- $A_i = 0$ *for all* $i \in \{1, \ldots, d-1\}$,

- $\sum_{v=0}^{n} A_v K_w(v) \geq 0$.

*Then any linear code $\mathcal{C} \subset \mathbb{F}_q^n$ of minimum distance $d$ is such that*

$$|\mathcal{C}| \leq \sum_{i=0}^{n} A_i.$$

Unfortunately, there is no closed formula for this bound, as the name says: the bound has to be programmed.

We may explain it though:

- the first condition makes sure that there is the zero codeword,
- the second condition makes sure the weight enumerators are non-negative,
- the third condition ensures that $d_H(\mathcal{C}) \geq d$,
- and the fourth condition ensures that the hypothetical dual code also has non-negative weight enumerators.

- The *weight distribution* of a code is given by all its weight enumerators $(A_i)_{0 \leq i \leq n}$.
- The *MacWilliams identities* show that the weight distribution of the dual code is completely determined given the weight distribution of the code.
- The *linear programming bound* uses the MacWilliams identities in its linear constraints to maximize the size of a code of a given minimum distance.

# 12 Rank-Metric Codes

# References


[1] L. Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.

[2] S. Ball. On sets of vectors of a finite vector space in which every subset of basis size is a basis. *Journal of the European Mathematical Society (EMS Publishing)*, 14(3), 2012.

[3] E. Berlekamp, R. McEliece, and H. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[4] G. D. Forney. Concatenated codes. 1965.

[5] V. D. Goppa. A new class of linear error-correcting codes. *Probl. Inf. Transm.*, 6:300–304, 1970.

[6] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 28–37. IEEE, 1998.

[7] R. W. Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[8] W. C. Huffman and V. Pless. *Fundamentals of error-correcting codes*. Cambridge university press, 2010.

[9] H. Lefmann, K. T. Phelps, and V. Rödl. Rigid linear binary codes. *Journal of Combinatorial Theory, Series A*, 63(1):110–128, 1993.

[10] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

[11] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\mathcal{O}(2^{0.054n})$. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 107–124. Springer, 2011.

[12] A. May and I. Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 203–228. Springer, 2015.

[13] R. J. McEliece. A public-key cryptosystem based On algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, Jan. 1978.

[14] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[15] R. Roth. *Introduction to coding theory*. Cambridge University Press, 2006.

[16] B. Segre. Curve razionali normali ek-archi negli spazi finiti. *Annali di Matematica Pura ed Applicata*, 39:357–379, 1955.

[17] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[18] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of complexity*, 13(1):180–193, 1997.

[19] J. H. Van Lint. *Introduction to coding theory*, volume 86. Springer Science & Business Media, 2012.

[20] J. Wood. The structure of linear codes of constant weight. *Transactions of the American Mathematical Society*, 354(3):1007–1026, 2002.