Chapter 1

Linear Codes

1

1

1 **Linear Codes**

1 Linear Codes

In the first chapter, we introduce the basic definitions, methods and results from the theory of error-correcting linear codes and its applications.

1.1 Introduction

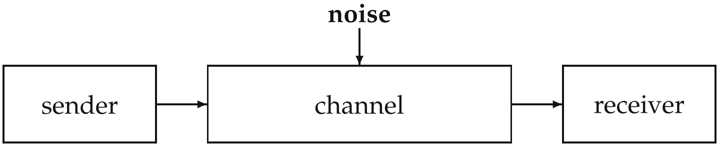
1.1

As Claude Shannon, the founding father of modern Information Theory puts it in [178],

> *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point."*

Error-correcting codes are used to improve the reliability of such communication systems. We may think of communication across large distances such as spacecraft communication, or basically any form of information transmission, including playing back a piece of music which was recorded previously and stored on some media, for instance. In any case, the goal is to transmit and reproduce the information as accurately as possible, even under unfavorable circumstances, like in an error-prone environment.

In order to make a mathematical approach possible, we introduce the following communication model. It has a sender and a receiver and they are supposed to communicate in one direction, so that information passes from the sender to the receiver. Thus, we suppose that between the sender and the receiver there is a *communication system*, a *channel*, and all information passes through this directed channel:



This channel may be unreliable, e.g. we are expecting that information may be altered as it is passing through. Often, this is called a *noisy channel*, appealing to the common experience that in a noisy room full of people it is usually impossible to understand a word someone has said at the other end of the room. Of course, we will make assumptions about the behavior of the channel, and it is clear that the channel should not be too bad, i.e. we require that a certain amount of information passes through correctly. Hence, we assume that the output of the channel is a more or less damaged version of the original input.

We suppose, for example, that the length of the signals, or *codewords* as we call them, that are sent through the channel is never changed by noise, and that, if noise inflicts a codeword, a letter is changed into another letter with the same probability for each letter. Such channels are called *symmetric*. On the receiving end, a process which is called *error-correction* takes place. Given the altered or damaged codeword, one tries to recover the original one by correcting errors. Of course, this is a difficult task as it is not clear where the error may have occurred (or if an error has occurred at all).

On the other end of the channel, the sender is trying to help by manipulating the messages *before they are transmitted*. This can be done by adding redundancy, for example, by repeating the message. The purpose of this is to protect the message, so that the influence of noise can later be corrected up to a certain degree.

A *message* is defined to be a finite sequence of elements of a given alphabet. Subsequently, such a sequence is also referred to as a *word.* There is no restriction in assuming that all messages are of a fixed length, say k. If the message is very long, we may break it up into pieces, and each such piece may be considered a message by itself. Hence without loss of generality, we assume that the messages are of size k. To enable error-correction, a process called *encoding* takes place. Here, we replace the message words by possibly longer sequences over the same alphabet, the codewords. The added redundancy will later enable the receiver to correct errors which may have occurred during transmission. The only requirement at this point is that the encoding map shall be injective, for otherwise the receiver would not be able to decide which message was sent, even under the most favorable circumstances when no error has occurred during transmission. It is customary to denote the length of codewords by n. The process of correcting errors and obtaining back the message is called *decoding*. This refined communication model is depicted in Fig. 1.1.

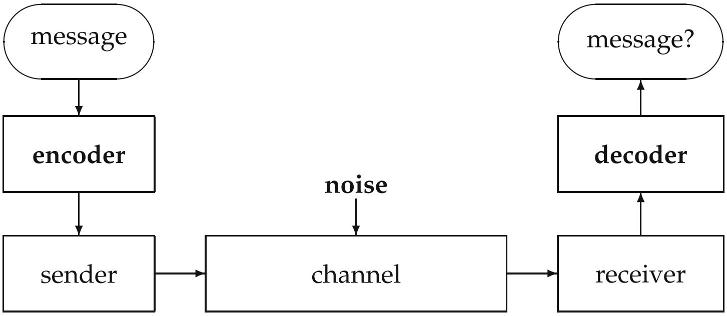


Fig. 1.1 The refined communication system

The arrows indicate that the communication is one-way. In particular, the possibility of asking for retransmission shall be excluded. Hence, error *correction* will be the main topic while error *detection* will be less important. As an aside, it should be mentioned that there are indeed codes which are based on the idea that bidirectional communication is possible. An important example is the well-known *ISBN-code,* which assigns to each book a unique number which may be used to identify the book. This number is composed of a certain number of digits. One digit plays the special role of a check digit. This check digit allows the detection of single errors and of interchanges of adjacent digits – a very frequent mistake. Of course, the idea is that the message can be repeated once the receiver has flagged the first transmission as erroneous. See Exercises 1.1.1–1.1.3 for details on this and similar codes. For further reading, see [98], [99], and [47].

Here is a very easy example that demonstrates the *metric principle* which is used for error detection and error correction. Suppose that we want to transmit over a noisy channel a message which is just one of the two answers "yes" or "no". These two messages can be identified with the one-element sequences 1 for "yes" and 0 for "no". So, $k = 1$ in this case, and the alphabet consists of the two symbols 0 and 1. A particularly simple way of adding redundancy is to use the well-known pedagogical principle of repetition of the message in question, or, in other words, to use the *repetition code.* Assume we have agreed to use three-fold repetition, i.e. $n = 3 \cdot k = 3 \cdot 1 = 3$. Then we intend to send as codewords sequences which are either 111 (for "yes") or 000 (for "no"). Thus, the *code* in question is the set $\{000, 111\}$. Moreover, we assume that this fact is known to the receiver. It motivates the following strategy:

- If neither 111 nor 000 but another sequence of length 3 is received, then the transmission has been distorted and there must be at least one error in the received sequence.

Hence, to begin with, we see that this repetition code is able to *detect* a certain amount of errors, in our particular example one or two. This means that in these cases we are certain that the transmission of the message is erroneous. But we should be aware of the fact that in the case of three errors the receiver cannot detect them. Moreover, the receiver can *correct* the received word into the original one, provided that not too may errors have occurred.

- If the receiver obtains one of the sequences 110, 101, 011, 100, 010, or 001, and if in addition we *assume that there was only one symbol changed by noise,* then we can simply decode the received word into the codeword which is *most similar,* i.e. into 111 in the first three cases, and into 000 in the latter three cases.

Hence, using the three-fold repetition code we are in a position to *correct a single error.* Our strategy is based on the fact that the messages that have to be sent, namely the sequences 111 or 000, differ in *three* places. The other six possibly received sequences differ from the original sequences in either one or two places. In fact, for each of the $2^3 = 8$ sequences which can be received, there is *exactly one* of the two vectors 111 and 000 which is *most similar*.

In order to demonstrate this principle in more detail, let us see what happens if we use four-fold repetition. Besides the correct sequences 1111 and 0000 there are fourteen other ones that can be received when errors have occurred. In this case we are in a slightly better position than with three-fold repetition: We can detect up to three errors (to be exact, we realize that errors have occurred), but we are not able to correct more than one error. The error correcting property has *not* improved. The reason is that in the case of two errors the received sequence consists of two letters 1 and two letters 0. In this situation we are unable to tell which codeword has been sent. In the case of five-fold repetition we can recognize up to four errors, and we can correct at most *two* errors, and so on.

The *metric principle* used is based on the fact that all the sequences that can occur at the receiver side differ from the correct sequences $1\dots1$ and $0\dots0$, of length n, in at most $\lfloor n/2 \rfloor$ many places. The number of places (or coordinates) in which two codewords differ is called the *Hamming distance* of the two codewords in question. The least Hamming distance between any two distinct codewords is called the *minimum distance* of the code. If d is the minimum distance of a linear code – not necessarily a repetition code – then up to $t := \lfloor (d-1)/2 \rfloor$ errors can be corrected, while up to $d-1$ errors are recognized. That is, if no more than $d-1$ errors occur, we detect that something is wrong. Later on, we will make this more precise.

The quality of a code with messages of length k and codewords of length n is indicated by

- the quotient k/n, the *information rate* of the code, which measures the effort needed for the transmission of an encoded message,
- the *relative minimum distance* d/n which gives roughly twice the proportion of errors that can be corrected in each encoded message (it is also called the *error-correction rate*),
- the *complexity of the encoding and of the decoding procedure*.

Summarizing, the main goal of Coding Theory is to provide codes with high information rate, high error correction rate and with a low complexity of encoding and decoding.

An important and intensely studied class of codes are the *linear codes*. These are just the k-dimensional subspaces of an n-dimensional vector space over a finite field. They form a subclass of the more general class of *block codes,* which are merely subsets of an n-dimensional space. The structure of linear codes can be analyzed using methods of Linear Algebra and Algebra as well as Combinatorics and Geometry.

In this introductory chapter, our goal is to discuss the fundamentals of the theory of linear codes. We also classify linear codes according to their error-correcting qualities. Codes with similar metric structure are collected into *isometry classes* of codes. Finally, we present an algorithm to determine the minimum distance of a given linear code. In later chapters we will deepen the theory, the construction, and the generation of linear codes and their application, and we will describe some important families of codes.

Exercises

Exercise The ISBN-code ("International Standard Book Number") is a sequence of ten elements $x_{10}, \ldots, x_1$ taken from the set $\{0, 1, \ldots, 9, X\}$. This sequence is divided into four parts of variable length, which must be separated by hyphens or spaces. The hyphens or spaces increase the readability and indicate the borders between the four different parts. These characters, however, do not influence the ability of the code to detect and correct errors. **E.1.1.1**

1. The first subsequence $x_{10}, \ldots$ (mostly of length 1) encodes the language, or, rather, the language region in which the book was printed. 0 stands for English speaking countries, 3 for the German speaking ones.

2. The next subsequence encodes the publishing company. It consists of at least two entries.

3. The sequence of the following entries, $\ldots, x_2$, is a number chosen by the publisher to identify the book.

The entries $x_{10}, \ldots, x_2$ are taken from the set $\{0, 1, \ldots, 9\}$.

4. The final entry, x_1, is the residue modulo 11 of $-\sum_{i=2}^{10} x_i \cdot i$. If this residue happens to be equal to 10, one puts $x_1 := X$.

Show that this code recognizes a single error as well as an interchange of two neighboring entries, and that it allows the reconstruction of an unreadable entry.

E.1.1.2 **Exercise** The ISSN-code ("International Standard Serial Number") is a sequence of eight elements $x_8, \ldots, x_1$ taken from the set $\{0, 1, \ldots, 9, X\}$. This sequence is divided into two parts, each consisting of four digits, which must be separated by a hyphen. Similar to the ISBN-code, the entries $x_8, \ldots, x_2$ are taken from the set $\{0, 1, \ldots, 9\}$, and the final entry x_1 is determined such that $\sum_{i=1}^{8} x_i \cdot i \equiv 0 \bmod 11$ is satisfied. If $x_1 = 10$, then x_1 is represented as X. This code has exactly the same properties as the ISBN-code.

1. Determine the ISSN-number of the Bayreuther Mathematische Schriften from the sequence `ISSN 0172-?062` where the digit x_4 is not readable.
2. The number `ISSN 0174-1062` is not a valid ISSN-number. Assuming that exactly one error occurred, determine all valid ISSN-numbers which could be represented by the given one.

E.1.1.3 **Exercise** The EAN-code ("European Article Number") has two basic formats, the 8 and 13 digit variants. The 13 digit code is more common, so we discuss it here. The 8 digit code is generally used where space is restricted. The EAN code is intended as a world wide standard (although some countries use other systems), therefore, no two products may have the same EAN number. To ease the administration of number allocation, each country using EAN has a country identifier at the start of the code. For instance the digits 00 to 13 identify the USA and Canada, 40 to 44 Germany, and 90 to 91 Austria. Other countries have 2 or 3 digit prefixes. The rest of the EAN code is divided into the manufacturer number which can be of variable length, the item reference number, assigned by the manufacturer, and the check digit. In general, both the manufacturer number and the item reference number consist of 5 digits. This means that in this case a manufacturer can have up to 10^5 products. For that reason, those manufacturers which produce a smaller number of products get longer manufacturer codes. The check digit is the last number. All 13 digits $x_{13}, \ldots, x_1$ are taken from the set $\{0, 1, \ldots, 9\}$. The check digit x_1 is determined by the other digits such that

$$\sum_{i \equiv 1 \bmod 2} x_i + 3 \cdot \sum_{i \equiv 0 \bmod 2} x_i \equiv 0 \bmod 10$$

is satisfied.

1. Show that the EAN-code recognizes a single error and allows the reconstruction of an unreadable entry, but in general it does not detect a swap of two neighboring entries.
2. The EAN of books can easily be obtained from their ISBN-number. As prefix, the three digits 978 are used, regardless of the country in which the

book was published. Then the ISBN-number with the check digit stripped is appended. Finally, the EAN check digit is computed from these 12 digits as described above. Compute the EAN-code of the present book!

The EAN is also coded in a machine-readable version as a barcode. For this purpose, the EAN is encoded as a binary sequence of bars and spaces. A 1 in the code is represented by a *bar section* and a 0 by a *space section.* Consecutive 1′s or 0′s are combined to form wider bars or spaces. The EAN barcode consists of the following parts:

- The left-hand starting section of the form 101,
- binary encodings of the digits $x_{12}, \ldots, x_7$,
- the center pattern of the form 01010,
- binary encodings of the digits $x_6, \ldots, x_1$,
- the right-hand closing section of the form 101.

For the encoding of $x_{12}, \ldots, x_1$ three different codes are used, codes A, B, and C. (See also [104, 1.2.5 Beispiel].) The codes A and B are applied for encoding $x_{12}, \ldots, x_7$, and code C is used for encoding $x_6, \ldots, x_1$. So far the digit x_{13} has not been encoded. Depending on the value of x_{13}, different sequences of the codes A and B are applied for the encoding of $x_{12}, \ldots, x_7$. They are given in the table below:

x_{13}	x_{12}	x_{11}	x_{10}	x_9	x_8	x_7
0	A	A	A	A	A	A
1	A	A	B	A	B	B
2	A	A	B	B	A	B
3	A	A	B	B	B	A
4	A	B	A	A	B	B
5	A	B	B	A	A	B
6	A	B	B	B	A	A
7	A	B	A	B	A	B
8	A	B	A	B	B	A
9	A	B	B	A	B	A

digit	code A	code B	code C
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

We realize that for encoding x_{12}, code A is always used. If $x_{13} = 0$ then all digits $x_{12}, \ldots, x_7$ are encoded with code A. In all other cases, codes A and B are each used to encode three digits.

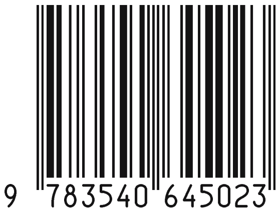
The three codes A, B, and C encode each digit as a binary word of length 7. Each codeword consists of two bar and two space sections. No bar or space is longer than four elements. All codewords of codes A and B start with 0 and end with 1. All codewords of code C start with 1 and end with 0. Actually, it would be enough to describe the codewords of code A. In order to obtain the codewords of code C from code A, exchange the 0's and 1's in the codewords of A. In order to obtain the codewords of code B from code C, reverse the order of each codeword of C.

Moreover, we realize that no codeword occurs in two different codes, and no codeword of A is the reverse of a codeword in C. This fact, together with the rule that x_{12} is always encoded with code A allows the determination of the direction (from left to right or from right to left) in which a barcode is read. When reading from left to right, after the left-hand starting section 101, the reader comes across an element of code A. When reading from right to left, after the reverse of the right-hand closing section, which is again 101, the reader comes across the reverse of an element of code C. Consequently, after reading the first digit it is possible to determine the direction of reading.

Finally, let us consider the following example. The book *Codierungstheorie, Springer, Berlin, 1998,* by some of the present authors and K.-H. Zimmermann, has the `ISBN 3-540-64502-0`. First, this number is encoded as an EAN of the form 9783540645023 where the last 3 is the EAN check digit. Since $x_{13} = 9$, we have to apply the codes $ABBABA$ for the encoding of $x_{12}, \dots, x_7$. This way we obtain the following binary representation of the bar code of 9783540645023.

101	left-hand starting
0111011 0001001 0100001 0110001 0011101 0001101	$x_{12} \dots x_7$
01010	center pattern
1010000 1011100 1001110 1110010 1101100 1000010	$x_6 \dots x_1$
101	right-hand closing

This gives a barcode of the form:



1.2 Linear Codes, Encoding and Decoding

As we have seen, the goal of Coding Theory is to provide methods which improve the reliability of communication via a noisy channel. For example, we may think of the transmission of satellite photos taken in space and sent back to earth. For this purpose, one decomposes the picture into a large number of pixels (which stands for "picture elements"), each pixel having a certain grey value, for example. The grey value is then mapped to a number, which in turn is converted to a binary sequence by means of the binary representation of an integer (i.e. one of 0, 1, 10, 11, 100, 101, 110, 111 etc.). Hence, the messages, i.e. the grey values of the pixels, can be considered as *words* over the *alphabet* $\{0, 1\}$.

For example, in the case of six values of grey, we have the messages

$$0,\ 1,\ 10,\ 11,\ 100,\ 101.$$

By padding with zeros up-front, we can make the sequences all have length 3. We can also add the remaining sequences of that length over the given alphabet, which in our case gives

$$000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111.$$

The reader certainly knows that the two elements 0 and 1 are the elements of a field $\mathbb{F}$, the binary field. The above sequences can be considered as the vectors of $\mathbb{F}^3$, the three-dimensional vector space over $\mathbb{F}$,

$$\mathbb{F}^3 = \{000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111\}.$$

This vector space is called the *message space.*

In more general situations it will be a k-dimensional vector space $\mathbb{F}^k$ over a finite field $\mathbb{F}$, which may be different from the field of two elements, of course. Later on we will see that the order defines a field up to isomorphism. Therefore, a field consisting of q elements is indicated by $\mathbb{F}_q$. Moreover, it will turn out that the orders q of finite fields are exactly the prime powers $q = p^m$, p a prime and $m \in \mathbb{N}^* := \mathbb{N} \setminus \{0\}$. For example, for each prime number p, the integers $0, 1, 2, \ldots, p-1$ form the field $\mathbb{F}_p$ with respect to addition and multiplication modulo p. In the case when the original finite alphabet A does not consist of elements of a finite field, then we simply take a suitable finite field $\mathbb{F}$ with at least $|A|$ elements and rename the letters of A by elements of $\mathbb{F}$.

As we have seen, the encoding map should be injective. This means that we are looking for an *embedding* of $\mathbb{F}^k$ into some space $\mathbb{F}^n$, where $n \geq k$. In order to add redundancy, we usually let n be strictly larger than k. The encoding of messages is done using an *encoder*

$$\gamma \colon \mathbb{F}^k \to \mathbb{F}^n,$$

an injective linear mapping of the message space $\mathbb{F}^k$ into $\mathbb{F}^n$. For example, we may simply repeat the messages twice, which yields the following embedding of $\mathbb{F}^3$ into $\mathbb{F}^6$:

$$\gamma(\mathbb{F}^3) = \{000000, 001001, \ldots, 110110, 111111\} \subset \mathbb{F}^6.$$

1.2.1 **Definition (linear codes, generator matrices)** The image

$$C = \gamma(\mathbb{F}^k)$$

of the encoder γ is a subspace of $\mathbb{F}^n$ which is isomorphic to the message space $\mathbb{F}^k$. We call C a *linear* (n,k)*-code* or briefly an (n,k)*-code* over $\mathbb{F}$. The number k is its *dimension* and the number n will be called the *block length* or simply the *length* of the code C. The vectors in C are the *codewords* or *codevectors*.

The encoder can be expressed as multiplication by a matrix Γ of rank k. Using the *row convention*, i.e. by writing vectors as row-vectors, Γ turns out to be a $k \times n$-matrix. The embedding is then given by the map

$$\gamma\colon \mathbb{F}^k \to \mathbb{F}^n \ : \ v \mapsto v \cdot \Gamma,$$

and we obtain that

$$C = \gamma(\mathbb{F}^k) = \{v \cdot \Gamma \mid v \in \mathbb{F}^k\}.$$

For this reason, the matrix Γ, which is in general not uniquely determined, is called a *generator matrix* of C. Its rows form a *basis* of C. ◇

1.2.2 **Example** If $k = 1$ and $\mathbb{F}^1 = \{0, 1\}$ is the message space, then the *three-fold repetition code* $C = \{000, 111\}$, which is an embedding of $\mathbb{F}^1$ into $\mathbb{F}^3$, has the generator matrix

$$\Gamma = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}.$$

In this particular case, the generator matrix is uniquely determined, but this is exceptional. For example, in the case of $k = n = 3$, each regular 3×3-matrix over $\mathbb{F}$ is a generator matrix. ◇

Hence, there are usually plenty of generator matrices of a given code C, and it is clear from Linear Algebra that they can be obtained from Γ by applications of invertible matrices:

1.2.3 **Theorem** *The set of all generator matrices of a linear code with generator matrix* Γ *is*

$$\{B \cdot \Gamma \mid B \in \mathrm{GL}_k(\mathbb{F})\},$$

where $\mathrm{GL}_k(\mathbb{F})$ *is the set of all regular* $k \times k$*-matrices over* $\mathbb{F}$. □

Now we describe a way of considering $\mathbb{F}^n$ as a *metric space* in order to justify the metric principle used in the decoding process as described in the introduction. Usually we indicate vectors $v \in \mathbb{F}^n$ in the form

$$v = (v_0, v_1, \ldots, v_{n-1}).$$

Throughout the book we consistently use the recursive definition of natural numbers as *sets,*

$$n := \{0, \ldots, n-1\}, \text{ if } n > 0,\ 0 := \emptyset.$$

Thus, a vector v can be considered as a mapping from this *set* n to $\mathbb{F}$, with v_i the image of $i \in n$ under v,

$$v \,:\, n \to \mathbb{F} \,:\, i \mapsto v_i.$$

In this sense, the vector space can be identified with a set of mappings:

$$\mathbb{F}^n = \{v \mid v\colon n \to \mathbb{F}\}\,.$$

Of course, we also use the natural number n as the *cardinality* of sets of this order, but it should be always clear from the context if n means a set or a cardinality of a set.

The metric principle which we are going to describe is based on the following fact:

Definition and Theorem (Hamming metric) *The function* **1.2.4**

$$d\colon \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{N} \,:\, (u,v) \mapsto |\{i \mid i \in n,\ u_i \neq v_i\}|$$

is a metric on the vector space $\mathbb{F}^n$*, the* Hamming metric. *This means that the function d satisfies*

$$\begin{aligned} d(u,v) &= 0 \iff u = v, \\ d(u,v) &= d(v,u), \\ d(u,v) &\leq d(u,w) + d(w,v), \end{aligned}$$

for all $u, v, w \in \mathbb{F}^n$*. The nonnegative integer* $d(u,v)$ *is called the* Hamming distance *between the vectors* $u, v \in \mathbb{F}^n$*. Hence, the pair* $(\mathbb{F}^n, d)$ *is a metric space, the* Hamming space *of dimension n over* $\mathbb{F}$*. It will also be denoted by*

$$H(n,\mathbb{F}) \quad \textit{or by} \quad H(n,q),$$

if $\mathbb{F} = \mathbb{F}_q$*. The Hamming distance is invariant under translation and multiplication by nonzero scalars: For* $u, v, w \in H(n,\mathbb{F})$ *and* $\lambda \in \mathbb{F}, \lambda \neq 0$,

$$d(u,v) = d(u+w, v+w), \quad \textit{and} \quad d(u,v) = d(\lambda u, \lambda v).$$

Proof: The equations $d(u,v) = 0$ and $u = v$ are obviously equivalent, and the symmetry of d is trivial. To show the triangle inequality

$$d(u,v) \leq d(u,w) + d(w,v),$$

we only note that the i-th component contributes to the left hand side if and only if $u_i \neq v_i$, in which case it also contributes to the sum on the right hand side, since then $u_i \neq w_i$ or $v_i \neq w_i$. The invariance under translation and scalar multiplication follows from $u_i = v_i \iff u_i + w_i = v_i + w_i$ and from $u_i = v_i \iff \lambda u_i = \lambda v_i$. This completes the proof. □

A measure for the error correction capabilities of a linear code C is the least Hamming distance between two distinct codewords. The reason is that this value determines the *packing radius* of C, which is the largest integer t such that the balls of radius t centered at codewords are all disjoint (intuitively, we can "pack" the balls).

1.2.5 **Definition (minimum distance)** If C denotes a linear code, then its *minimum distance* is defined as

$$\text{dist}(C) := \min\{d(c,c') \mid c,c' \in C,\ c \neq c'\}. \qquad \diamond$$

A glance at Fig. 1.2 shows that the packing radius is the greatest integer which is strictly less than half the value of the minimum distance.

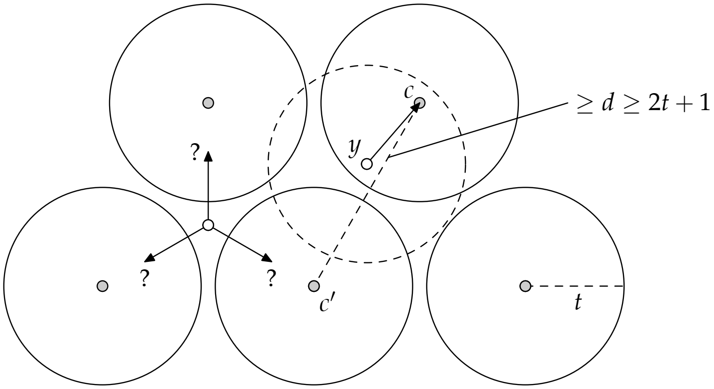


Fig. 1.2 The maximum-likelihood-decoding method

1.2.6 **Corollary** *The packing radius of a linear code C is* $\lfloor(\text{dist}(C) - 1)/2\rfloor$. □

The crucial point is the following decoding method, which is based on the concept of the packing radius:

Maximum-likelihood-decoding *It is possible to correct up to* **1.2.7**

$$t := \lfloor (\mathrm{dist}(C) - 1)/2 \rfloor$$

errors in the following way (cf. Fig. 1.2):

- *Using* maximum-likelihood-decoding, *a vector $y \in \mathbb{F}^n$ is decoded into a codeword $c \in C$ which is closest to y with respect to the Hamming metric. In formal terms: y is decoded into a codeword $c \in C$, such that*

  $$d(c,y) \leq d(c',y), \text{ for all } c' \in C.$$

  If there are several $c \in C$ with this property, one of them is chosen at random.

- *If the codeword $c \in C$ was sent and no more than t errors have occurred during transmission, the received vector is*

  $$y = c + e \in \mathbb{F}^n,$$

  where e denotes the error vector. *It satisfies*

  $$d(c,y) = d(e,0) \leq t,$$

  and hence c is the unique *element of C which lies in a ball of radius t around y. A maximum likelihood decoder yields this element c, and so we obtain the correct codeword.* □

We mention that codes of dimension $0 < k = n$ obviously have minimum distance $d = 1$, and soon we will see that in the case $k = n - 1$ we have $d \leq 2$.

If we want to *evaluate* the minimum distance of a code, we can, of course, evaluate the distances $d(c,c')$ of all

$$\binom{|C|}{2} = \binom{|\mathbb{F}|^k}{2}$$

unordered pairs of different codewords. But this is a very inefficient way to do so. A better approach is the following. For a vector v, we denote by

$$\mathrm{wt}(v) := d(v,0),$$

the *Hamming weight* of v. It is just the number of components of v which are nonzero. For a code C, the *minimum weight* of C is defined as

$$\min\{\mathrm{wt}(c) \mid c \in C,\ c \neq 0\},$$

and it is not difficult to show (Exercise 1.2.8) that, because of linearity, the following is valid:

1.2.8 **Corollary** *The minimum distance of linear codes is the minimum weight:*

$$\operatorname{dist}(C) = \min\{\operatorname{wt}(c) \mid c \in C \setminus \{0\}\}. \qquad \square$$

An (n,k)-code C with minimum distance d is called an (n,k,d)*-code* or a linear code of *type* (n,k,d). If C is an (n,k,d)-code over a field with q elements, we also say that it is an (n,k,d,q)-code.

1.2.9 **Corollary** *Using an (n,k,d)-code in connection with maximum-likelihood-decoding, we can correct up to*

$$t := \lfloor (d-1)/2 \rfloor$$

*errors. For this reason, (n,k,d)-codes are sometimes called t-*error-correcting *linear codes, for $t := \lfloor (d-1)/2 \rfloor$. Moreover, such a code is $(d-1)$-*error-detecting *since a word which was received with at least one and at most $d-1$ errors can never be another codeword.* □

This is of course the reason why the minimum distance of a code is of such importance.

We are now able to refine our communication model. Denoting by m a message, and by M the message space $\mathbb{F}^k$, we are faced with the situation of Fig. 1.3.

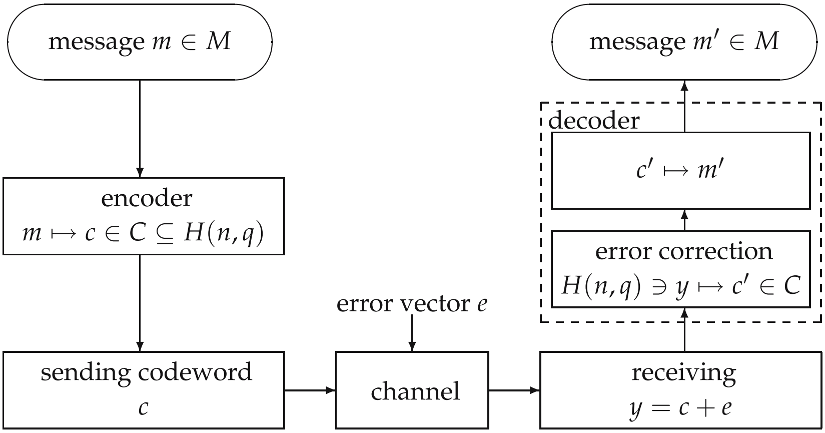


Fig. 1.3 The refined communication system once again

Of course, it is not always true that the message m' after decoding is equal to the message m which was sent originally. The main point is that maximum likelihood decoding ensures that $m = m'$, provided that $\operatorname{wt}(e) \leq \lfloor (d-1)/2 \rfloor$.

Exercises

Exercise Assume that V and W are two finite dimensional vector spaces over $\mathbb{F}$ with bases **E.1.2.1**

$$\mathcal{B} = (b^{(0)}, \dots, b^{(k-1)}) \text{ and } \mathcal{C} = (c^{(0)}, \dots, c^{(n-1)}).$$

Prove the following: Every vector space homomorphism $\varphi\colon V \to W$ is uniquely determined by its values on a basis of V. Assume that

$$\varphi(b^{(i)}) = \sum_{j \in n} \kappa_{ij} c^{(j)}, \quad \kappa_{ij} \in \mathbb{F},$$

for $i \in k$. If we collect the elements κ_{ij} in form of a matrix we obtain the *matrix representation*

$$M_{\mathcal{B},\mathcal{C}}(\varphi) := (\kappa_{ij}) = \begin{pmatrix} \kappa_{00} & \kappa_{01} & \cdots & \kappa_{0,n-1} \\ \kappa_{10} & \kappa_{11} & \cdots & \kappa_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \kappa_{k-1,0} & \kappa_{k-1,1} & \cdots & \kappa_{k-1,n-1} \end{pmatrix}$$

of φ with respect to the bases $\mathcal{B}$ and $\mathcal{C}$. Conversely, prove that any $k \times n$-matrix (κ_{ij}) over $\mathbb{F}$ determines a vector space homomorphism $\varphi\colon V \to W$ such that $M_{\mathcal{B},\mathcal{C}}(\varphi) = (\kappa_{ij})$. The homomorphism φ is given by

$$\varphi\Big(\sum_{i \in k} v_i b^{(i)}\Big) = \sum_{j \in n} w_j c^{(j)}$$

with

$$(w_0, \dots, w_{n-1}) = (v_0, \dots, v_{k-1}) \cdot (\kappa_{ij}).$$

This shows that a generator matrix Γ of an (n,k)-code over $\mathbb{F}$ describes a vector space homomorphism $\varphi\colon \mathbb{F}^k \to \mathbb{F}^n$.

In particular, if $\mathcal{B} = (b^{(0)}, \dots, b^{(k-1)})$ is a basis of V, every endomorphism of V can be represented as a $k \times k$-matrix over $\mathbb{F}$ with respect to this basis.

Exercise Let V and W be two finite dimensional vector spaces over $\mathbb{F}$ of dimension k and n respectively. Show that a homomorphism $\varphi\colon V \to W$ is injective if and only if $\dim(\varphi(V)) = \dim(V)$. Hence, the rows of any matrix representation (κ_{ij}) of φ are linearly independent, and the rank of (κ_{ij}) equals k, the number of its rows. Moreover, $\varphi\colon V \to W$ is an isomorphism if and only if φ is injective and $n = k$. **E.1.2.2**

E.1.2.3 **Exercise** Assume that V is a k-dimensional vector space over $\mathbb{F}$ with basis $\mathcal{B} = (b^{(0)}, \dots, b^{(k-1)})$. The matrix representation $M_{\mathcal{B},\mathcal{B}}(\varphi)$ of any automorphism φ of V is a $k \times k$-matrix.

1. Show that the rank of this matrix equals k, which means that it is a *regular* matrix.
2. Conversely, prove that any regular $k \times k$-matrix over $\mathbb{F}$ determines an automorphism of V.
3. Prove that $M_{\mathcal{B},\mathcal{B}}(\varphi_1\varphi_2)$ equals the product $M_{\mathcal{B},\mathcal{B}}(\varphi_2) \cdot M_{\mathcal{B},\mathcal{B}}(\varphi_1)$ for all automorphisms $\varphi_1, \varphi_2 \in \mathrm{Aut}(V)$.
4. Deduce from the previous result that the set $\mathrm{GL}_k(\mathbb{F})$ of all regular $k \times k$-matrices over $\mathbb{F}$ forms a group with respect to matrix multiplication, the *general linear group*.
5. Show that the mapping $\theta\colon \mathrm{Aut}(V) \to \mathrm{GL}_k(\mathbb{F})$ which maps a vector space automorphism φ of V to $M^{\top}_{\mathcal{B},\mathcal{B}}(\varphi)$, the transpose of its matrix representation, is a group isomorphism.
6. Changing the basis of V from $\mathcal{B} = (b^{(0)}, \dots, b^{(k-1)})$ to $\mathcal{C} = (c^{(0)}, \dots, c^{(k-1)})$ is described by the matrix representation $M_{\mathcal{B},\mathcal{C}}(\mathrm{id})$ of the identity mapping. This is also a regular quadratic matrix. Express $M_{\mathcal{C},\mathcal{C}}(\varphi)$ in terms of these matrices.

E.1.2.4 **Exercise** Prove 1.2.3. Hint: Use the fact that any encoding γ of a linear code C is a vector space isomorphism from $\mathbb{F}^k$ to C.

E.1.2.5 **Exercise** Which code has an invertible generator matrix?

E.1.2.6 **Exercise** Show that in a linear code over $\mathbb{F} = \{0, 1\}$ either all codewords begin with 0, or exactly half of them begin with 0 and half of them begin with 1.

E.1.2.7 **Exercise** Give a formal proof of 1.2.6.

E.1.2.8 **Exercise** Give a formal proof of 1.2.8.

E.1.2.9 **Exercise** Assume that G is an abelian group which contains a subset A with the following three properties:

1. If $a_1, a_2 \in A$, then $a_1 - a_2 \in A$,

2. if $b_1, b_2 \in G \setminus A$, then $b_1 - b_2 \in A$,
3. if $a \in A$ and $b \in G \setminus A$, then $a + b \notin A$.

Show that $A = G$ or A is a proper subset of G and there is an element b_0 of $G \setminus A$ such that $G = A \cup (b_0 + A)$ where $b_0 + A = \{b_0 + a \mid a \in A\}$. (This exercise generalizes the last two exercises.)

Exercise E.1.2.10

1. Show that there are $\binom{n}{m}$ binary vectors of length n and weight m.
2. Show that there are $(q-1)^m\binom{n}{m}$ vectors in $\mathbb{F}^n$ of weight m, provided that $\mathbb{F}$ consists of q elements.
3. For $x \in H(n,q)$, how many vectors $y \in H(n,q)$ satisfy $d(x,y) \leq m$?

Exercise Let u and v be binary vectors of length n with $d(u,v) = d$. For $r, s \in \mathbb{N}$ determine z by E.1.2.11

$$z := \left|\{w \in \{0,1\}^n \mid d(u,w) = r \text{ and } d(v,w) = s\}\right|.$$

Prove the following statements: If $d + r - s \geq 0$ and $d + r - s$ is even then

$$z = \binom{d}{i}\binom{n-d}{r-i},$$

where $i = (d + r - s)/2$. If $d + r - s$ is odd or $d + r - s < 0$, then $z = 0$. If $r + s = d$, then $z = \binom{d}{r}$.

Exercise Let u, v and w be binary vectors which are pairwise at distance d. Show that d is even and that there exists exactly one vector which is at distance $d/2$ from u, v, w. If u, v, w and x are binary vectors which are pairwise at distance d, show that there exists at most one vector at distance $d/2$ from u, v, w and x. E.1.2.12

Exercise Show that if C is a binary linear code, and $a \in \{0,1\}^n \setminus C$, then $C \cup (a + C)$ is also a linear code. E.1.2.13

Exercise Define the “intersection” of two binary vectors u and v to be the vector E.1.2.14

$$u \wedge v := (u_0v_0, \ldots, u_{n-1}v_{n-1})$$

which has ones only where both u and v have ones. Also, let

$$u \vee v := (1 - (1 - u_0)(1 - v_0), \ldots, 1 - (1 - u_{n-1})(1 - v_{n-1}))$$

be the “union” of u and v, i.e. the vector which is one if at least one of u or v is one. Show that

$$\operatorname{wt}(u+v) = \operatorname{wt}(u) + \operatorname{wt}(v) - 2\operatorname{wt}(u \wedge v) = \operatorname{wt}(u \vee v) - \operatorname{wt}(u \wedge v).$$

1.3

1.3 Check Matrices and the Dual Code

Let us now address the important issue of decoding. It turns out that Linear Algebra helps to understand the problem quite a bit. We will discuss a decoding method using what is called the *coset leader algorithm*. Nevertheless, this problem is computationally hard and may only be practical for small parameters. However, it illustrates some very important concepts of Coding Theory which are also useful for other purposes, too.

An (n,k)-code $C \subseteq H(n,\mathbb{F})$ can be considered both as the *image of an injective linear mapping* $\gamma\colon \mathbb{F}^k \to \mathbb{F}^n$, and as the *kernel of a surjective linear mapping* $\delta\colon \mathbb{F}^n \to \mathbb{F}^{n-k}$ (Exercise 1.3.1). This leads to the following

1.3.1 **Definition (check matrices)** Let C be an (n,k)-code over $\mathbb{F}$. There exists an $(n-k) \times n$-matrix Δ over $\mathbb{F}$ which is of rank $n-k$ and satisfies

$$C = \ker(\delta) = \{w \in \mathbb{F}^n \mid w \cdot \Delta^\top = 0\},$$

where $\Delta^\top$ denotes the transpose of the matrix Δ. Any such matrix is called *check matrix* of C. ◇

Codes over the field $\mathbb{F}_2 := \{0,1\}$ of two elements are called *binary codes.* Codes over the field $\mathbb{F}_3 := \{0,1,2\}$ of three elements are called *ternary codes*, whereas codes over a four-element field $\mathbb{F}_4$ are called *quaternary.*

1.3.2 **Example** Consider the following check matrix over the field $\mathbb{F}_2 = \{0,1\}$ of two elements, consisting of a single row of length $n \geq 2$,

$$\Delta := \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}.$$

It is a check matrix of a binary $(n,n-1)$-code C. Each codeword

$$c = (c_0, \dots, c_{n-1}) \in C$$

is of even weight, since

$$0 = c \cdot \Delta^\top = c_0 + \ldots + c_{n-1} \equiv \operatorname{wt}(c) \mod 2.$$

Conversely, $w \cdot \Delta^\top = 0$ for each vector $w \in \mathbb{F}_2^n$ of even weight, i.e., C consists of the vectors of even weight in $\mathbb{F}_2^n$, and so C has minimum distance $d = 2$. This shows that C can detect one error. It is called a *parity check code,* since C can be obtained in the following way: Take $C' := \mathbb{F}_2^{n-1}$ as the message space and add to each of its elements $c' = (c_0, \ldots, c_{n-2})$ a further coordinate c_{n-1}, a single bit called a *parity check bit,* given by

$$c_{n-1} := \begin{cases} 1 & \text{if } \mathrm{wt}(c') \text{ is odd,} \\ 0 & \text{otherwise.} \end{cases}$$

The purpose of the parity check bit is to ensure that each codeword of the extended code C has even weight. A generator matrix of C is

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

◇

Let us abbreviate the all-one vector $(1, \ldots, 1)$ as $\mathbf{1}$ and the vector whose entries are all zero by $\mathbf{0}$. We also write $\mathbf{1}_n$ or $\mathbf{0}_n$ for such vectors of length n. For instance, the check matrix and the generator matrix of the above example can be written as

$$\Delta = (\ \mathbf{1}_n\ ) \text{ and } \Gamma = (\ I_{n-1} \mid \mathbf{1}_{n-1}^\top\ ),$$

respectively, where I_{n-1} indicates the identity matrix of rank $n-1$.

Now we introduce for each linear code C another code which is closely related to C via its check and its generator matrices. Using the *standard bilinear form*

$$\langle w, w' \rangle := \sum_{i \in n} w_i w'_i \ \in \ \mathbb{F},$$

we associate with C the following subspace:

Definition (the dual code, self-orthogonal and self-dual codes) The *dual code* to $C \subseteq H(n, \mathbb{F})$ is defined to be the space of vectors that are orthogonal to C with respect to the standard bilinear form: **1.3.3**

$$C^\perp := \{w \in \mathbb{F}^n \mid \forall\, c \in C : \langle c, w \rangle = 0\}.$$

A code C is called *self-orthogonal* if $C \subseteq C^\perp$ and we say that it is *self-dual* if $C = C^\perp$. ◇

The standard bilinear form has the following property:

$$\langle w, w' \rangle = 0, \text{ for all } w \in \mathbb{F}^n \iff w' = 0 \in \mathbb{F}^n.$$

For $v \in \mathbb{F}^k$, $w \in \mathbb{F}^n$ and a generator matrix Γ of C it follows from

$$\langle v \cdot \Gamma, w \rangle = \langle v, w \cdot \Gamma^\top \rangle$$

that

$$C^\perp = \{w \in \mathbb{F}^n \mid w \cdot \Gamma^\top = 0\}.$$

This shows that the generator matrix Γ of C is a check matrix of $C^\perp$. Consequently, $C^\perp$ is a linear $(n, n-k)$-code. Since $(C^\perp)^\perp = C$ (cf. Exercise 1.3.12), the converse is true as well, and we obtain

1.3.4 **Corollary** *The check matrices of a code C are the generator matrices of the dual code $C^\perp$ and vice versa. Dually, the check matrices of the dual code are the generator matrices of the code.* □

It is now time to present an example of a linear code which can correct one error. This is everyone's first code which is not a repetition code or any of the other trivial examples. It was introduced by Hamming. Before we define this code, let us make one more definition.

1.3.5 **Definition** Let $b \geq 2$ be an integer. Every nonnegative integer $m \leq b^k - 1$ can be expressed in the form

$$m = \sum_{i \in k} a_i b^i, \quad \text{where} \quad 0 \leq a_i < b, \ \text{for } i = 0, 1, \dots, k-1.$$

We call this the base b representation of m. The a_i are called the *digits* in the representation and we write

$$m = (a_{k-1}, \dots, a_1, a_0)_b.$$

The integer b is called *base*. The expression is unique up to the number of leading zeros (we do not distinguish between two such representations which only differ in the number of leading zeros). The case $b = 10$ is of course the usual representation of integers in decimal, whereas $b = 2$ gives us the binary numbers. Notice the ubiquitous reverse ordering of the digits with respect to the index set k. ◇

The announced code is described in the following

1.3.6 **Example** Consider the binary representations of the numbers from 1 to 7, $(0,0,1)_2$, $(0,1,0)_2$, $(0,1,1)_2$, $(1,0,0)_2$, $(1,0,1)_2$, $(1,1,0)_2$, and $(1,1,1)_2$, respectively. We may form the binary matrix

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

whose columns are exactly these binary representations (slightly "mixed up" however). We may take Δ to be the check matrix of a binary code of length 7. The rowspace of Δ is the dual space of the code, and hence the code is the set of vectors c with $c \cdot \Delta^\top = 0$. Using Linear Algebra, we can find a basis for this 4-dimensional space, and writing the basis vectors in the rows of a matrix we find that the code is generated by

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

This is the $(7,4)$-Hamming-code. Actually, it is a member of a whole class of codes which are all called *Hamming-codes*. The more general definition of a Hamming-code will follow at the beginning of Chapter 2. By enumerating the 16 codewords and counting weights, one can easily determine that the minimum distance of this code is 3. Note that it cannot be larger than that since we see vectors of weight 3 in the rows of the generator matrix Γ. However, we need to convince ourselves that there is no word of *lower* weight. Hence this code has type $(n,k,d,q) = (7,4,3,2)$. By 1.2.7, it is a 1-*error correcting code.* Its information rate is $k/n = 4/7 \approx 0,57$. By comparison, the 1-error correcting repetition code of length 3 has information rate $1/3 \approx 0,33$. This is already a good improvement. ◇

Using check matrices, we can easily formulate an important decoding procedure which will turn out to agree with maximum-likelihood-decoding. For this purpose, we recall the definition of a *coset* of C, which is a subset of $\mathbb{F}^n$ of the form

$$a + C := \{a + c \mid c \in C\},$$

where a is an element of $\mathbb{F}^n$. It is possible to decompose $\mathbb{F}^n$ into pairwise disjoint cosets of C (cf. Exercise 1.3.4),

$$\mathbb{F}^n = \bigcup_i (a^{(i)} + C).$$

As coset representatives we use *coset leaders* $a^{(i)}$, which are elements of smallest weight in their coset,

$$\mathrm{wt}(a^{(i)}) \le \mathrm{wt}(a^{(i)} + c), \quad \text{for all } c \in C.$$

The decoding algorithm itself can be described as follows:

1.3.7 **Syndrome decoding** Let Δ be a check matrix of C and suppose that the Hamming space $H(n,\mathbb{F}) \supseteq C$ is decomposed into cosets $a^{(i)}+C$ such that the chosen representatives $a^{(i)}$ are coset leaders. For each vector $w \in \mathbb{F}^n$ we call the vector

$$w \cdot \Delta^\top$$

its *syndrome.* Assume that the vector y has been received. To determine the coset $a^{(i)}+C$ containing y we proceed as follows:

- If $y \in a^{(i)}+C$, say $y = a^{(i)}+c$, then

$$y \cdot \Delta^\top = (a^{(i)}+c)\cdot\Delta^\top = a^{(i)}\cdot\Delta^\top + c\cdot\Delta^\top = a^{(i)}\cdot\Delta^\top,$$

  i.e. the received vector y has the same syndrome as its coset leader.

- Syndromes of different $a^{(i)}$ are distinct, since

$$a^{(i)}\cdot\Delta^\top = a^{(j)}\cdot\Delta^\top \Rightarrow (a^{(i)}-a^{(j)})\cdot\Delta^\top = 0 \Rightarrow a^{(i)}-a^{(j)} \in C \Rightarrow i=j.$$

  Consequently, we can deduce the coset number i from the syndrome of y by comparing it to the (pairwise distinct!) syndromes of the coset leaders.

- Having the coset number i of y and its coset leader $a^{(i)}$ at hand, we simply subtract $a^{(i)}$ from y in order to obtain a codeword c. This is called the *syndrome decoding method:*

$$y \mapsto c := y - a^{(i)}.$$

  For short: *Subtract from the received vector its coset leader!*

In fact, this is the maximum-likelihood-decoding method, since $y = a^{(i)}+c$ implies that

$$d(y,c) = \mathrm{wt}(y-c) = \mathrm{wt}(a^{(i)}),$$

as we have seen already. Therefore, since $a^{(i)}$ is a leader, c is one of the codewords next to y. ◇

1.3.8 **Example** Consider the check matrix

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

of a binary $(6,3)$-code. The following table presents the coset leaders and the corresponding syndromes:

$a^{(i)}$	$a^{(i)} \cdot \Delta^\top$
000000	000
000001	001
000010	010
000100	100
100000	110
000101	101
001000	011
010000	111

The reader should carefully note that coset leaders are usually not uniquely determined. In our example there are *several coset leaders* admissible for the syndrome 101. They are 000101, 101000 and 010010. ◇

The following theorem provides an important characterization of the minimum distance in terms of check matrices. Remember that the check matrix is not unique. The statement holds true for any check matrix.

Theorem *The check matrix Δ of an (n, k, d)-code over $\mathbb{F}$ with $0 < k < n$ has the following properties:* **1.3.9**

1. *Δ is an $(n-k) \times n$-matrix over $\mathbb{F}$ of rank $n-k$,*
2. *any $d-1$ columns are linearly independent, and*
3. *there exist d columns that are linearly dependent.*

Conversely, any matrix Δ satisfying these properties is a check matrix of an (n, k, d)-code over $\mathbb{F}$.

Proof: To begin with, we assume that Δ is a check matrix of such a code. By 1.3.4, Δ is a generator matrix of the dual code, i.e. an $(n-k) \times n$-matrix over $\mathbb{F}$ of rank $n-k$. Now let c be a word in C of minimum weight d. Then $c \cdot \Delta^\top = 0$, since the rows of Δ are a basis for $C^\perp$. But this means that there is a nontrivial linear combination of d columns of Δ that gives the zero vector (namely, the columns corresponding to the nonzero entries of c). Moreover, since there is *no* codeword $c \neq 0$ with Hamming weight strictly less than d, any $d-1$ columns of Δ are linearly independent.

Conversely, if we are given such a matrix Δ, the rank condition implies that the set

$$\{w \in \mathbb{F}^n \mid w \cdot \Delta^\top = 0\}$$

is a subspace of dimension k. Moreover, as before, we find that it is a code of minimum distance d. □

From this result we deduce the following criterion which can be used in many cases:

1.3.10 **Corollary** *Each $(n-k)\times n$-matrix over $\mathbb{F}$ of rank $n-k$ with the property that any $d-1$ of its columns are linearly independent is a check matrix of a linear (n,k)-code C over $\mathbb{F}$ with minimum distance* $\mathrm{dist}(C)\geq d$*, for short: of an $(n,k,\geq d)$-code.* □

The excluded case when $k=n$ is obviously trivial, since in this case $d=1$.

Exercises

E.1.3.1 **Exercise** Check that in fact any (n,k)-code C can be described as the kernel of a surjective linear mapping from $\mathbb{F}^n$ to $\mathbb{F}^{n-k}$. Hint: Assume that Γ is a generator matrix of C. Let $\{b^{(0)},\ldots,b^{(n-k-1)}\}$ be a basis of the solution space of the homogeneous linear system $\Gamma\cdot x^\top=0$, where $x\in\mathbb{F}^n$. Then C is the kernel of the mapping $\mathbb{F}^n\to\mathbb{F}^{n-k}\ :\ w\mapsto w\cdot\Delta^\top$ with

$$\Delta=\left(\begin{array}{c} b^{(0)} \\ \hline \vdots \\ \hline b^{(n-k-1)} \end{array}\right),$$

an $(n-k)\times n$-matrix over $\mathbb{F}$. Thus, Δ is a check matrix of C.

E.1.3.2 **Exercise** List all codewords of the binary codes C_0 and C_1 with the check matrices

$$\Delta_0=\begin{pmatrix}1&1&0&0\\1&0&1&1\end{pmatrix}\text{ and }\Delta_1=\begin{pmatrix}0&1&1&1\\1&0&1&1\end{pmatrix}.$$

How are these two codes related?

E.1.3.3 **Exercise** Assume that Δ is the check matrix of a linear code C. Describe the set of all check matrices of C.

E.1.3.4 **Exercise** Verify that $\mathbb{F}^n$ is the union of pairwise disjoint cosets of C.

E.1.3.5 **Exercise**

1. Check that the rowspace of the matrix Γ in 1.3.6 is indeed the dual space of the rowspace of Δ.
2. Verify the claim about the minimum distance of the $(7,4)$ Hamming-code made in 1.3.6.

Exercise Prove that for a binary code with check matrix Δ, the syndrome is the transpose of the sum of the columns of Δ where the errors have occurred. **E.1.3.6**

Exercise Compute coset leaders for the binary code generated by **E.1.3.7**

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

Decode the vectors $(1,1,0,1,0,0)$ and $(1,1,1,1,1,1)$ using the method of 1.3.7.

Exercise Evaluate the minimum distances of the binary codes which are generated by **E.1.3.8**

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Exercise Let C be an (n,k)-code. Consider the block matrix $\Gamma = (I_k \mid A)$, where A is a $k \times (n-k)$-matrix and I_k denotes the unit matrix. Show that Γ is a generator matrix of C if and only if $\Delta = (-A^\top \mid I_{n-k})$ is a check matrix of C. **E.1.3.9**

Exercise Prove that for each generator matrix Γ and every check matrix Δ of C the products $\Gamma \cdot \Delta^\top$ and $\Delta \cdot \Gamma^\top$ are zero matrices. **E.1.3.10**

Exercise Over any finite field $\mathbb{F}$, the $(n, n-1)$ parity check code C is obtained from the message space $\mathbb{F}^{n-1}$ by adding a parity check bit $c_n := -\sum_{i=0}^{n-1} c_i$. Find a generator matrix for this code and determine the minimum distance. **E.1.3.11**

Exercise Verify that $(C^\perp)^\perp = C$. **E.1.3.12**

Exercise Assume that C and C' are linear codes of length n and let $C + C' := \{c + c' \mid c \in C,\ c' \in C'\}$. Show that $(C + C')^\perp = C^\perp \cap C'^\perp$. **E.1.3.13**

Exercise A linear code C is self-orthogonal if and only if $\langle c, c' \rangle = 0$ for all $c, c' \in C$. Show that C is self-dual if and only if C is self-orthogonal and C is of dimension $k = n/2$ (and hence n is even). **E.1.3.14**

Exercise Construct binary self-dual codes of lengths 4 and 8. **E.1.3.15**

E.1.3.16 **Exercise** Let C be a binary, self-orthogonal code.

1. Show that each word of C is even and that $C^\perp$ contains the all-one vector $\mathbf{1}$.
2. Assume in addition that the length n of C is odd and that the dimension of C is $(n-1)/2$. Show that
$$C^\perp = C \cup (\mathbf{1} + C).$$

E.1.3.17 **Exercise** Show that a code with check matrix $\Delta = (I_k \mid A)$ is self-dual if and only if A is a square matrix with $A \cdot A^\top = -I_k$.

E.1.3.18 **Exercise** Show the following:

1. If $u, v \in \mathbb{F}_2^n$, then $\langle u, v\rangle \equiv \mathrm{wt}(u \wedge v) \bmod 2$ (where $u \wedge v$ is as in Exercise 1.2.14).
2. If $u \in \mathbb{F}_2^n$, then $\langle u, u\rangle \equiv \mathrm{wt}(u) \bmod 2$.
3. If $u \in \mathbb{F}_3^n$, then $\langle u, u\rangle \equiv \mathrm{wt}(u) \bmod 3$.

E.1.3.19 **Exercise** If C is a binary, self-orthogonal code, show that every codeword has even weight. Furthermore, if each row of the generator matrix Γ of C has weight divisible by 4, then so does every codeword.

E.1.3.20 **Exercise** Let C be a ternary, self-orthogonal code. Show that $\mathrm{wt}(c) \equiv 0 \bmod 3$ for every codeword $c \in C$.

E.1.3.21 **Exercise** Let C be a code whose generator matrix Γ has the property that no column of Γ is zero and no two columns of Γ are linearly dependent. Show that $\mathrm{dist}(C^\perp) \geq 3$. (Such codes will be called *projective* in 6.1.14.)

1.4

1.4 Classification by Isometry

As we have seen, the coding theoretic properties of a code depend primarily on the Hamming distances between different codewords and between codewords and non-codewords. For example, the closest pair of codewords determines the error-correction rate of a code. Moreover, it may be that one code can be mapped onto another by means of a map which preserves the Hamming distances. Clearly, in any practical application, one code would be as good as the other, as far as error-correction is concerned. It seems natural to call such codes equivalent. In this section we study a corresponding notion of equivalence, by means of which codes can be classified.

Of course, only the types of essentially distinct – i.e. nonequivalent – codes are of interest. In fact, there are various ways in which such an equivalence relation can be defined. We discuss three such relations. These relations are indeed only refinements of each other, meaning that there is one relation which is strongest. The other relations are "weaker" in the sense that codes which are equivalent under the strongest relation may be inequivalent under the other two relations. The three relations are motivated by concepts from Projective Geometry, see Section 3.7 for more on that.

Recall that an *equivalence relation* R on a set X is a subset of $X \times X$ such that for all $x, y, z \in X$ we have

- $(x, x) \in R$ (reflexivity),
- $(x, y) \in R$ if and only if $(y, x) \in R$ (symmetry),
- $(x, y), (y, z) \in R$ implies that also $(x, z) \in R$ (transitivity).

The equivalence class of $x \in X$ with respect to R is the set

$$[x]_R := \{y \in X \mid (x, y) \in R\},$$

and the set of all equivalence classes with respect to R is indicated as X/R. It forms a decomposition of X into pairwise disjoint and nonempty subsets. Instead of $(x, y) \in R$ we usually write $x \sim y$ where $\sim$ denotes the equivalence relation.

Two (n, k)-codes $C, C' \subseteq H(n, q)$ are of the same quality if there exists a mapping

$$\iota\colon H(n, q) \to H(n, q)$$

with $\iota(C) = C'$ which preserves the Hamming distance, i.e.

$$d(w, w') = d(\iota(w), \iota(w')), \text{ for all } w, w' \in H(n, q).$$

Mappings with the latter property are called *isometries*. Using this notion we introduce the following concept which is in fact *the central concept of the present book:*

Definition (isometric codes) Two linear codes $C, C' \subseteq H(n, q)$ are called *isometric* if there exists an isometry of $H(n, q)$ that maps C onto C'. ◇ **1.4.1**

Obvious isometries are the permutations of the coordinates. These isometries will be called *permutational isometries*. Recall that the set of bijections from a set X to itself forms a group, the *symmetric group*

$$S_X := \{\pi \mid \pi\colon X \to X,\ \pi \text{ is bijective}\}.$$

The multiplication in this group is the composition of mappings,

$$(\pi \circ \rho)(x) := \pi(\rho(x)).$$

We write S_n for the symmetric group on the set $X = n = \{0, \ldots, n-1\}$.

1.4.2 **Definition (permutationally isometric codes)** Two linear codes $C, C' \subseteq H(n,q)$ are *permutationally isometric* if there exists a permutational isometry of $H(n,q)$ that maps C onto C'. This means that there is a permutation π in the symmetric group S_n such that

$$C' = \pi(C) = \{\pi(c) \mid c \in C\}, \text{ and } d(c, \widetilde{c}) = d(\pi(c), \pi(\widetilde{c})),$$

for all $c, \widetilde{c} \in C$, where

$$\pi(c) = \pi(c_0, \ldots, c_{n-1}) := (c_{\pi^{-1}(0)}, \ldots, c_{\pi^{-1}(n-1)}). \qquad \diamond$$

Isometries which are also linear mappings are called *linear isometries* (with respect to the Hamming metric). Linear isometries leave the Hamming weight invariant, since by linearity we have $\iota(0) = 0$, and therefore also

$$\text{wt}(v) = d(v, 0) = d(\iota(v), \iota(0)) = d(\iota(v), 0) = \text{wt}(\iota(v)).$$

1.4.3 **Definition (linearly isometric codes)** Two linear codes $C, C' \subseteq H(n,q)$ are *linearly isometric* if there exists a linear isometry of $H(n,q)$ that maps C onto C'. ◇

We remark that what we call linearly isometric is often called isometric (unqualified) or monomially isometric in the literature. Our reason for calling it linearly isometric is two-fold. First, we will see shortly that this, together with the special case of permutational isometry, is not the only way in which codes can be isometric. Secondly, concerning the notion of equivalence, we felt that the concept of monomial mapping is not that well-known. Hence we chose to make reference to the fact that these isometries are induced by linear mappings.

We might have imposed a seemingly weaker condition by asking for the existence of a *local* linear isometry between C and C' only, i.e. an isometry of C and not necessarily of $H(n,q)$, that maps C onto C'. It can be shown, see

6.8.4, that each such local linear isometry can be extended to a linear isometry of $H(n,q)$. Later on we will see that the isometry relation is an equivalence relation on the set of codes with block length n over $\mathbb{F}_q$, and in later chapters we will consider the corresponding isometry classes in detail.

In order to characterize linear isometries, we have to study linear maps of the vector space $\mathbb{F}_q^n$ and investigate their effect on the Hamming distance. That is, we study linear maps of $H(n,q)$. Recall that any linear map is defined by the images of the unit vectors. Since linear isometries preserve the Hamming weight, a unit vector $e^{(i)}$ is mapped to a nonzero multiple of a unit vector, i.e.

$$\iota(e^{(i)}) = \kappa_j e^{(j)}, \text{ for suitable } j \in n,\ \kappa_j \in \mathbb{F}_q^* := \mathbb{F}_q \setminus \{0\}\,.$$

Moreover, the sum of two *different* unit vectors is of weight 2, and so different unit vectors are mapped under ι to nonzero multiples of different unit vectors. Hence, there exists a unique permutation π in the symmetric group S_n and a unique mapping φ from $n = \{0, \dots, n-1\}$ to $\mathbb{F}_q^*$ such that

$$\iota(e^{(i)}) = \varphi(\pi(i))e^{(\pi(i))}.$$

Therefore, we may record ι as a pair of mappings,

$$\iota = (\varphi;\pi).$$

In terms of these mappings, applying ι to $v := \sum_{i\in n} v_i e^{(i)}$ gives

$$\iota(v) = (\varphi;\pi)(v) = \sum_{i\in n} v_i \varphi(\pi(i))e^{(\pi(i))} = \sum_{i\in n} \varphi(i) v_{\pi^{-1}(i)} e^{(i)},$$

i.e.

$$(\varphi;\pi)((v_0, \dots, v_{n-1})) = (\varphi(0)v_{\pi^{-1}(0)}, \dots, \varphi(n-1)v_{\pi^{-1}(n-1)}).$$

Using matrix multiplication, we could also write

$$(\varphi;\pi)((v_0, \dots, v_{n-1})) = (v_0, \dots, v_{n-1}) \cdot M_{(\varphi;\pi)}^{\top},$$

where $M_{(\varphi;\pi)}$ is the matrix whose k-th column is zero except for the (i,k)-entry which is $\varphi(i)$. Here $i = \pi(k)$, so that

$$M_{(\varphi;\pi)} := \begin{array}{cc} \begin{array}{ccccccc} & & & k & & & \end{array} & \\ \left(\begin{array}{ccccccc} & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \\ 0 & \dots & 0 & \varphi(i) & 0 & \dots & 0 \\ & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \end{array}\right) & i = \pi(k). \end{array}$$

Conversely, any linear mapping with

$$e^{(i)} \longmapsto \varphi(\pi(i))e^{(\pi(i))},$$

for $\varphi\colon n \longrightarrow \mathbb{F}_q^*$ and $\pi \in S_n$, is a linear isometry. Moreover, linear isometries are invertible, and the composition of two of them is again a linear isometry. A straightforward calculation shows (Exercise 1.4.1) that

$$(\psi;\rho)((\varphi;\pi)(v)) = (\psi\varphi_\rho;\rho\pi)(v), \qquad v \in H(n,q),$$

where $\psi\varphi_\rho(i) := \psi(i)\varphi(\rho^{-1}(i))$. Summarizing we obtain

1.4.4 **Corollary** *The linear isometries form the group*

$$\left\{(\varphi;\pi) \,\middle|\, \varphi\colon n \to \mathbb{F}_q^*,\ \pi \in S_n\right\},$$

called the group of linear isometries *of the Hamming space. Multiplication in this group is given by the formula*

$$(\psi;\rho)(\varphi;\pi) = (\psi\varphi_\rho;\rho\pi).$$

The matrices representing the elements of this group form

$$M_n(q) := \left\{M_{(\varphi;\pi)} \,\middle|\, \varphi\colon n \to \mathbb{F}_q^*,\ \pi \in S_n\right\},$$

and they multiply according to the rule

$$M_{(\psi;\rho)} \cdot M_{(\varphi;\pi)} = M_{(\psi\varphi_\rho;\rho\pi)}.$$

The correspondence between a linear map and the associated matrix with respect to a fixed basis constitutes the isomorphism

$$(\varphi;\pi) \longmapsto M_{(\varphi;\pi)}$$

between these two groups. □

The application of the linear isometry group to the Hamming space is our central concept, and it is a special case of the general notion of group action which we will use in other situations, too. Hence, we carefully introduce the basic definitions and results on group actions at this point.

Actions of groups on sets play an important role in Algebra, in Combinatorics, in Topology, but also in the sciences (Chemistry, Computer Science and Physics, in particular). For more details on group actions we refer the reader to [108].

An *action* of a group G (which we assume to be written multiplicatively) from the left on a nonempty set X is defined by a mapping

$$G \times X \to X \ : \ (g, x) \mapsto gx$$

with the properties

$$(gg')x = g(g'x) \ \text{ and } \ 1x = x,$$

for $x \in X$, $g, g' \in G$ and the identity element 1 of G. We abbreviate such an action of G on X from the left by

$${}_GX.$$

An equivalent characterization of a group action is as follows (Exercise 1.4.4).

Lemma *Let ${}_GX$ be a group action. Then the mapping* **1.4.5**

$$\delta \colon G \to S_X \ : \ g \mapsto \overline{g}, \textit{ where } \overline{g} \colon x \mapsto gx,$$

from G to the symmetric group S_X is a homomorphism. The kernel *of the action is by definition the kernel of this homomorphism, i.e. the set of group elements that fix each $x \in X$.* □

We call δ the *permutation representation* induced by the action of ${}_GX$, $\overline{g} = \delta(g)$ is the permutation *induced by g on X* and $\overline{G} := \delta(G)$ the permutation group *induced by G on X.* Actions from the right are defined similarly. In the following, we define the basic notions for actions from the left. It is clear that corresponding notions can be introduced for actions from the right as well.

The crucial point is that ${}_GX$ induces the following relation $\sim_G$ on X:

$$x \sim_G y \quad :\Longleftrightarrow \quad \exists g \in G \ : \ gx = y.$$

It is easy to prove that $\sim_G$ is indeed an *equivalence relation* on X (Exercise 1.4.4). The proof is based on the following fact which is immediate from the definition of group actions and which is of fundamental importance:

$$gx = x' \iff x = g^{-1}x'.$$

The equivalence class

$$G(x) = \{gx \mid g \in G\}$$

of $x \in X$ is called the *G-orbit* or, briefly, the *orbit* of x. We use the notation

$$G \backslash\backslash X := \{G(x) \mid x \in X\}$$

to denote the set of orbits of G on X. A minimal but complete set T of orbit representatives is called a *transversal* of the orbits. Since $\sim_G$ is an equivalence relation on X, $G\backslash\backslash X$ is a *set partition* of X, i.e. a complete dissection of X into pairwise disjoint and nonempty subsets $G(t)$, for $t \in T$:

1.4.6
$$X = \dot{\bigcup_{\omega \in G\backslash\backslash X}} \omega = \dot{\bigcup_{t \in T}} G(t).$$

Several basic examples of group actions appearing in Group Theory and Combinatorics are described in Exercises 1.4.5 to 1.4.7. It is easy to check that for any group action ${}_GX$ the orbits $G(x)$ and $\overline{G}(x)$, $x \in X$, coincide, whence, $G\backslash\backslash X = \overline{G}\backslash\backslash X$. A group action is called *finite* if both G and X are finite. If X is finite, then the action ${}_{\overline{G}}X$ is always finite.

We are now going to introduce an important action of a group on a set of mappings. This action will be the prototype action for the enumeration of isometry classes of codes later on. For nonempty sets X and Y, the set of mappings from X to Y is denoted as

$$Y^X := \{f \mid f\colon X \to Y\}.$$

If G acts on X, then we can define an action of G on Y^X as follows:

1.4.7
$$G \times Y^X \to Y^X \ : \ (g, f) \mapsto f \circ \overline{g}^{\,-1}.$$

Here $\overline{g}$ is the permutation induced by g on X as introduced in 1.4.5. Thus, under this action, we associate to the pair (g, f) the composition $f \circ \overline{g}^{\,-1}$, i.e. the mapping $\tilde{f} \in Y^X$ with $\tilde{f}(x) = f(g^{-1}x)$, for all $x \in X$.

Let us now introduce the wreath product of two groups. As it turns out, the linear isometry group of the Hamming space will be such a product.

1.4.8 **Definition (wreath product)** Consider an action ${}_GX$ and a group H. The *wreath product* of H with G, with respect to ${}_GX$, consists of the set

$$H \wr_X G := H^X \times G = \{(\varphi; g) \mid \varphi\colon X \to H,\ g \in G\},$$

with multiplication defined by

$$(\varphi; g)(\varphi'; g') := (\varphi\varphi'_g; gg'),$$

where $(\varphi\varphi'_g)(x) := \varphi(x) \cdot \varphi'_g(x)$ and $\varphi'_g(x) := \varphi'(g^{-1}x)$, for $x \in X$. The identity element is

$$1_{H \wr_X G} = (\epsilon; 1_G),$$

where $\epsilon \in H^X$ is the constant mapping $\epsilon\colon x \mapsto 1_H$, and $1_G, 1_H$ denote the identity elements of G and H, respectively. The inverse of $(\varphi; g) \in H \wr_X G$ is

$$(\varphi; g)^{-1} = (\varphi^{-1}_{g^{-1}}; g^{-1}),$$

where

$$\varphi^{-1}(x) := \varphi(x)^{-1} \text{ and } \varphi^{-1}_{g^{-1}} := (\varphi_{g^{-1}})^{-1} = (\varphi^{-1})_{g^{-1}}. \qquad \diamond$$

So, the wreath product $H \wr_X G$ comes together with an action of G on X. It may happen that the group H acts on another set Y, say. In this case, we can define an action of $H \wr_X G$ on the set of mappings Y^X in the following way.

$$H \wr_X G \times Y^X \to Y^X \ : \ ((\varphi; g), f) \mapsto \widetilde{f}, \text{ where } \widetilde{f}(x) := \varphi(x) f(g^{-1}x). \qquad \textbf{1.4.9}$$

This action is a host of further actions, some of which will be described next. These further actions are in fact actions of various subgroups of $H \wr_X G$ (cf. Exercise 1.4.5). The first case is when the group G is trivial and all mappings $\varphi : X \to H$ are constant. In this situation, only the group H acts on the set Y, such that the corresponding action on the set of functions Y^X is

$$H \times Y^X \to Y^X \ : \ (h, f) \mapsto \overline{h} \circ f. \qquad \textbf{1.4.10}$$

Another action is given by the direct product $H \times G$ of the groups H and G, which acts as follows:

$$(H \times G) \times Y^X \to Y^X \ : \ ((h, g), f) \mapsto \overline{h} \circ f \circ \overline{g}^{-1}. \qquad \textbf{1.4.11}$$

The purpose of Exercise 1.4.11 is to show that these definitions yield group actions. The action of the wreath product 1.4.9 is a generalization of 1.4.7, 1.4.10, and 1.4.11.

Example (the linear isometry group) Our paradigmatic example of an action as in 1.4.9 is the following one. Take as H the multiplicative group $\mathbb{F}_q^*$ of the field $\mathbb{F}_q$. Let G be the symmetric group S_n acting on the set $n = \{0, \ldots, n-1\}$. Thus **1.4.12**

$$H \wr_X G := \mathbb{F}_q^* \wr_n S_n = \left\{ (\varphi; \pi) \,\middle|\, \varphi \colon n \to \mathbb{F}_q^*,\ \pi \in S_n \right\}.$$

The action on $Y^X := \mathbb{F}_q^n$ is given in the following way:

$$\mathbb{F}_q^* \wr_n S_n \times \mathbb{F}_q^n \to \mathbb{F}_q^n \ : \ \big((\varphi; \pi), v\big) \mapsto \big(\varphi(0) v_{\pi^{-1}(0)}, \ldots, \varphi(n-1) v_{\pi^{-1}(n-1)}\big).$$

Equivalently, in terms of Linear Algebra, we could also write

$$M_n(q) \times H(n, q) \to H(n, q) \ : \ (M_{(\varphi;\pi)}, v) \mapsto v \cdot M^{\top}_{(\varphi;\pi)}.$$

Since $M_n(q) \simeq H \wr_n S_n$ is called the *full monomial group of degree n over H*, the group of linear isometries of the Hamming space is the full monomial group of degree n over the multiplicative group of the field. $\diamond$

We are now in a position to formulate linear isometry in terms of group actions.

1.4.13 **Remarks** Let us apply what we know about linear isometry groups, their actions on vector spaces and the general theory of group actions on sets of mappings Y^X. We iterate this process of constructing actions in the following way:

- We start from the action of the linear isometry group of $H(n,q)$,

$$ {}_{\mathbb{F}_q^* \wr_n S_n}\left(\mathbb{F}_q^n\right) = {}_{M_n(q)}\Big(H(n,q)\Big). $$

- Then we use the fact that the set of mappings

$$ 2^{H(n,q)} = \{F \colon H(n,q) \to \{0,1\}\} $$

  can be identified with the power set of $H(n,q)$ by identifying F with the inverse image $F^{-1}(\{1\})$ of 1, which is a subset of $H(n,q)$.

- The given action of the linear isometry group of $H(n,q)$ induces the action

$$ {}_{G}\left(Y^X\right) \;:=\; {}_{\mathbb{F}_q^* \wr_n S_n}\left(2^{\mathbb{F}_q^n}\right) \;=\; {}_{M_n(q)}\left(2^{H(n,q)}\right). $$

- Correspondingly, the orbits in

$$ M_n(q) \backslash\!\backslash\, 2^{H(n,q)} $$

  are the linear isometry classes of sub*sets* of $H(n,q)$ or *block codes*.

- Linear subspaces of $H(n,q)$ are of course also subsets of $H(n,q)$, and the previous remarks apply to them as well. It turns out that each element in the orbit of a linear subspace under the isometry group is again a linear subspace (this follows since the isometry group $M_n(q)$ is linear). Thus, these are the orbits we are interested in most. They are *the linear isometry classes of linear codes*.
  In later chapters we will enumerate these classes, construct representatives and provide a method for randomly generating subsets of $\mathbb{F}_q^n$ that are uniformly distributed over these classes. ◇

Next, we describe linear codes and their isometry classes as orbits under certain group actions by using results from the Exercises 1.4.14, 1.4.15, and 1.4.16, replacing the subspaces by generator matrices, i.e. by bases, so that they can be handled by a computer as well:

1.4.14 **Theorem**

1. *Assume that $\mathbb{F}_q^{k\times n,k}$ denotes the set of all $k \times n$ matrices of rank k over $\mathbb{F}_q$, $k \geq 1$, and $\mathrm{GL}_k(q)$ the set of all regular $k \times k$-matrices over $\mathbb{F}_q$. The set of all generator matrices of the linear (n,k)-code C with generator matrix $\Gamma \in \mathbb{F}_q^{k\times n,k}$ is the orbit*

$\mathrm{GL}_k(q)(\Gamma) = \{B \cdot \Gamma \mid B \in \mathrm{GL}_k(q)\}$. *Whence the set of all linear* (n,k)*-codes over* $\mathbb{F}_q$*, we indicate it as* $\mathcal{U}(n,k,q)$*, can be identified with*

$$\mathrm{GL}_k(q)\backslash\!\backslash\mathbb{F}_q^{k\times n,k}.$$

2. *The linear isometry group* $M_n(q)$ *acts on* $\mathcal{U}(n,k,q), k \geq 1,$ *according to*

$$M_n(q) \times \mathcal{U}(n,k,q) \to \mathcal{U}(n,k,q) \ : \ (M_{(\varphi;\pi)}, C) \mapsto \left\{ c \cdot M_{(\varphi;\pi)}^\top \;\middle|\; c \in C \right\}.$$

The linear isometry class of the linear (n,k)*-code* C *is the orbit*

$$M_n(q)(C).$$

Hence, the set of linear isometry classes of linear (n,k)*-codes is*

$$M_n(q)\backslash\!\backslash\mathcal{U}(n,k,q).$$

3. *The direct product* $\mathrm{GL}_k(q) \times M_n(q), k \geq 1,$ *acts on* $\mathbb{F}_q^{k\times n,k}$ *by*

$$\big(\mathrm{GL}_k(q) \times M_n(q)\big) \times \mathbb{F}_q^{k\times n,k} \to \mathbb{F}_q^{k\times n,k} \ : \ \big((B, M_{(\varphi;\pi)}), \Gamma\big) \mapsto B \cdot \Gamma \cdot M_{(\varphi;\pi)}^\top$$

and so the set of linear isometry classes of linear (n,k)*-codes corresponds to the set of orbits*

$$(\mathrm{GL}_k(q) \times M_n(q))\backslash\!\backslash\mathbb{F}_q^{k\times n,k}.$$

□

Exercises

Exercise Show that **E.1.4.1**

- linear isometries are invertible,
- the composition of two of them is again a linear isometry,
- the composition satisfies

$$(\psi;\rho)((\varphi;\pi)(v)) = (\psi\varphi_\rho;\rho\pi)(v), \qquad v \in H(n,q),$$

where $\psi\varphi_\rho(i) := \psi(i)\varphi(\rho^{-1}(i))$, and

- the representing matrices satisfy

$$M_{(\psi;\rho)} \cdot M_{(\varphi;\pi)} = M_{(\psi\varphi_\rho;\rho\pi)}.$$

Exercise Let U be a nonempty subset of a finite group G (written multiplicatively). Show that U is a subgroup if and only if U is closed under multiplication, i.e. **E.1.4.2**

$$u, u' \in U \Longrightarrow u \cdot u' \in U.$$

E.1.4.3 **Exercise** Verify 1.4.5.

E.1.4.4 **Exercise** Check that $\overline{g}$ is in fact a permutation and $\sim_G$ an equivalence relation.

E.1.4.5 **Exercise** If ${}_GX$ is a group action and U is a subgroup of G, prove that

$$U \times X \to X \,:\, (u, x) \mapsto ux$$

is a group action of U on X, the *restriction of* ${}_GX$ *to* U. Prove that each orbit $G(x)$ is a union of U-orbits.

E.1.4.6 **Exercise** If G is a group, prove that both

$$G \times G \to G \,:\, (g, x) \mapsto gx$$

and

$$G \times G \to G \,:\, (g, x) \mapsto xg^{-1}$$

are group actions of G on G. They are called the *left regular* or *right regular representation* of G, respectively. Prove that $G(x) = G$ for any $x \in G$. A group action with just one orbit is called *transitive*. Hence, the left regular and the right regular representation are transitive group actions.

Let U be a subgroup of G. Determine the orbits of the restricted action ${}_UG$. In the first case they are called *right cosets,* in the second case *left cosets* of U. Prove that all orbits $U(x)$ for $x \in G$ are of the same size. If G is a finite group, deduce that the order of U divides the order of G. This is *Lagrange's Theorem*.

E.1.4.7 **Exercise** Show that an action of G on a set X induces natural actions of G on $\binom{X}{k}$, the set of all k-subsets of X, for $0 \leq k \leq |X|$, and on 2^X, the power set of X, which is the set of all subsets of X. This natural action of g on the subset A of X is given by $(g, A) \mapsto \{gx \mid x \in A\}$.

E.1.4.8 **Exercise** Consider a group action ${}_GX$, a normal subgroup $U \trianglelefteq G$ and the restricted action ${}_UX$. Prove the following facts:

- For each orbit $U(x)$ and any $g \in G$, the set $gU(x)$ is again an orbit of U on X. Indeed $gU(x) = U(gx)$.
- The group G acts on the set $U\backslash\backslash X$ of the U-orbits by

$$G \times U\backslash\backslash X \to U\backslash\backslash X \,:\, (g, U(x)) \mapsto U(gx).$$

- The factor group G/U acts on the set $U\backslash\backslash X$ via
$$G/U \times U\backslash\backslash X \to U\backslash\backslash X \;:\; (gU, U(x)) \mapsto U(gx).$$
We call this action a *factor action* of G with respect to U and denote it by
$$_{G/U}(U\backslash\backslash X).$$
- Up to identification of the U-orbits with the set of their elements, the following equations hold:
$$G\backslash\backslash X = G\backslash\backslash(U\backslash\backslash X) \text{ and } G\backslash\backslash X = (G/U)\backslash\backslash(U\backslash\backslash X).$$

Exercise Use Exercise 1.4.8 in order to prove: An action of the direct product $H \times G$ on X induces both a natural action of H on the set of orbits of the restricted action $_GX$: **E.1.4.9**

$$H \times (G\backslash\backslash X) \to G\backslash\backslash X \;:\; (h, G(x)) \mapsto G(hx),$$

and a natural action of G on the orbits of the restricted action $_HX$:

$$G \times (H\backslash\backslash X) \to H\backslash\backslash X \;:\; (g, H(x)) \mapsto H(gx).$$

Show that the orbit of $G(x) \in G\backslash\backslash X$ under H is the set of orbits of G on X that form $(H \times G)(x)$, while the orbit of $H(x) \in H\backslash\backslash X$ under G consists of the orbits of H on X, that form $(H \times G)(x)$. Hence

$$(H \times G)(x) = \bigcup_{h\in H} G(hx) = \bigcup_{g\in G} H(gx).$$

Prove the following identity for a finite set X:

$$|H\backslash\backslash(G\backslash\backslash X)| = |G\backslash\backslash(H\backslash\backslash X)| = |(H \times G)\backslash\backslash X|.$$

Exercise Assume that both $_GX$ and $_HX$ are group actions with $g(hx) = h(gx)$ for all $g \in G$, $h \in H$, and $x \in X$. Prove that **E.1.4.10**

$$(H \times G) \times X \to X \;:\; ((h, g), x) \mapsto h(gx)$$

describes an action of the direct product $H \times G$ on X.

Exercise Assume that X and Y are sets and H is a group which acts on Y. Prove that 1.4.10 describes an action of H on Y^X. **E.1.4.11**

If, in addition, $_GX$ is another group action, then use Exercise 1.4.10 to show that 1.4.11 defines an action of $H \times G$ both on the domain and the range of these mappings. Note that $\overline{g}$ stands for the permutation representation of g acting on X, whereas $\overline{h}$ denotes the permutation representation of h acting on Y.

E.1.4.12 **Exercise** Let V be a vector space over $\mathbb{F}$. Show that the multiplicative group $\mathbb{F}^*$ acts on V by

$$\mathbb{F}^* \times V \to V \ : \ (\lambda, v) \mapsto \lambda v.$$

Prove that the orbit of 0 is of size one, and all the other orbits are of the same length. For $v \neq 0$ the orbit $\mathbb{F}^*(v)$ describes a punctured one-dimensional subspace of V, i.e. the subspace without the zero vector. If $\mathbb{F} = \mathbb{F}_q$, then the orbit of $v \neq 0$ is of size $q-1$.

E.1.4.13 **Exercise** Show that the group of regular $k \times k$-matrices over $\mathbb{F}$ acts on $\mathbb{F}^k$ by

$$\mathrm{GL}_k(\mathbb{F}) \times \mathbb{F}^k \to \mathbb{F}^k \ : \ (B, v) \mapsto \left(B \cdot v^\top\right)^\top = v \cdot B^\top.$$

Prove that the orbit of 0 is of size one. Moreover, show that this action commutes with the action of $\mathbb{F}^*$ described in Exercise 1.4.12, and deduce from Exercise 1.4.10 that the direct product $\mathrm{GL}_k(\mathbb{F}) \times \mathbb{F}^*$ acts on $\mathbb{F}^k$. Describe the orbits $(\mathrm{GL}_k(\mathbb{F}) \times \mathbb{F}^*)(v)$ with the methods of Exercise 1.4.9.

E.1.4.14 **Exercise** Let the set of $k \times n$-matrices over $\mathbb{F}_q$ be denoted by $\mathbb{F}_q^{k\times n}$, and the set of $k \times n$-matrices of rank r by $\mathbb{F}_q^{k\times n,r}$. Show that $\mathrm{GL}_k(q) := \mathrm{GL}_k(\mathbb{F}_q), k \geq 1$, acts both on $\mathbb{F}_q^{k\times n}$ and $\mathbb{F}_q^{k\times n,r}$ by

$$(B, \Gamma) \mapsto B \cdot \Gamma$$

where $B \in \mathrm{GL}_k(q)$ is a regular matrix, and Γ is a $k \times n$-matrix.

From 1.2.3 deduce that the orbit $\mathrm{GL}_k(q)(\Gamma)$ of $\Gamma \in \mathbb{F}_q^{k\times n,k}$ determines the set of all generator matrices of the code C with Γ. Thus the set of all linear (n,k)-codes over $\mathbb{F}_q$ can be identified with the set of orbits $\mathrm{GL}_k(q)\backslash\backslash\mathbb{F}_q^{k\times n,k}$.

E.1.4.15 **Exercise** Show that the full monomial group $M_n(q)$ acts on $\mathcal{U}(n,k,q)$ by

$$M_n(q) \times \mathcal{U}(n,k,q) \to \mathcal{U}(n,k,q) \ : \ (M_{(\varphi;\pi)}, C) \mapsto \left\{ c \cdot M_{(\varphi;\pi)}^\top \;\middle|\; c \in C \right\}.$$

E.1.4.16 **Exercise** Show that $M_n(q)$ acts both on $\mathbb{F}_q^{k\times n}$ and $\mathbb{F}_q^{k\times n,r}$ by

$$(M_{(\varphi;\pi)}, \Gamma) \mapsto \Gamma \cdot M_{(\varphi;\pi)}^\top$$

where $M_{(\varphi;\pi)} \in M_n(q)$ is a monomial matrix, and Γ is a $k \times n$-matrix. Moreover, show that this action commutes with the action of $\mathrm{GL}_k(q)$ described in Exercise 1.4.14 and thus deduce from Exercise 1.4.10 that the direct product $\mathrm{GL}_k(q) \times M_n(q)$ acts on $\mathbb{F}_q^{k\times n}$ and $\mathbb{F}_q^{k\times n,r}$. Describe $(\mathrm{GL}_k(q) \times M_n(q))(\Gamma)$ with the methods of Exercise 1.4.9.

From Exercise 1.4.14 deduce that for $\Gamma \in \mathbb{F}_q^{k\times n,k}$, a generator matrix of the (n,k)-code C, the orbit $(\mathrm{GL}_k(q) \times M_n(q))(\Gamma)$ consists of all generator matrices of codes which are linearly isometric to C. Therefore, the set of orbits $(\mathrm{GL}_k(q) \times M_n(q))\backslash\!\backslash \mathbb{F}_q^{k\times n,k}$ is in bijection to the linear isometry classes of linear (n,k)-codes over $\mathbb{F}_q$.

1.5 Semilinear Isometry Classes of Linear Codes 1.5

It is, of course, a legitimate question to ask for *generalizations* of the concept of linear isometry *by relaxing* the condition of *linearity.* The only requirement in addition to isometry will be that the admissible isometries map subspaces onto subspaces. To be more precise the image of a subspace under an isometry is again a subspace of $\mathbb{F}_q^n$. Under these assumptions we derive for $n \geq 3$ that these mappings preserve the dimension, i.e. they map (n,k)-codes to (n,k)-codes, and that they are the *semilinear* isometries of $\mathbb{F}_q^n$ (cf. 1.5.7). In order to prove this we need a more detailed analysis of isometries. At first we prove that it suffices to investigate isometries ι of $\mathbb{F}_q^n$ with $\iota(0) = 0$.

Lemma *If $\iota \colon \mathbb{F}_q^n \to \mathbb{F}_q^n$ is an isometry, then* 1.5.1

$$\iota' : \mathbb{F}_q^n \to \mathbb{F}_q^n \ : \ \iota'(v) := \iota(v) - \iota(0), \qquad v \in \mathbb{F}_q^n,$$

is again an isometry of $\mathbb{F}_q^n$ and $\iota'(0) = 0$.

Conversely, if $\iota' : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is an isometry with $\iota'(0) = 0$, then for any $w \in \mathbb{F}_q^n$ the mapping

$$\iota \colon \mathbb{F}_q^n \to \mathbb{F}_q^n \ : \ \iota(v) := \iota'(v) + w, \qquad v \in \mathbb{F}_q^n,$$

is an isometry with $\iota(0) = w$. □

This result, the proof of which is left to the reader as Exercise 1.5.1, shows that it suffices to consider only isometries ι with $\iota(0) = 0$. For example, if ι maps subspaces onto subspaces, then this condition always holds, since the null space $\{0\}$ is mapped onto $\{0\}$. If $\iota(0) = 0$, then ι also preserves the weight, since

$$\mathrm{wt}(\iota(v)) = d(\iota(v), 0) = d(\iota(v), \iota(0)) = d(v, 0) = \mathrm{wt}(v), \qquad v \in \mathbb{F}_q^n.$$

Lemma *Each isometry ι on a finite vector space $\mathbb{F}_q^n$ is bijective. If it satisfies $\iota(0) = 0$, then it permutes the orbits* 1.5.2

$$\mathbb{F}_q^*(e^{(i)}) = \{\kappa e^{(i)} \mid \kappa \in \mathbb{F}_q^*\}$$

of the unit vectors with respect to the action of $\mathbb{F}_q^*$ by left multiplication. In formal terms:

$$\exists\, \pi \in S_n \;\forall\, i \in n\colon\; \iota\,(\mathbb{F}_q^*(e^{(i)})) = \mathbb{F}_q^*(e^{(\pi(i))}).$$

Proof: 1. It is easy to see that ι is injective: $\iota(u) = \iota(v)$ implies

$$0 = d(\iota(u), \iota(v)) = d(u, v),$$

and so $u = v$. Since ι is a map from the *finite* set $\mathbb{F}_q^n$ to itself, it is also one-to-one.

2. Now we note that, for each $i \in n$ and $\lambda \in \mathbb{F}_q^*$, there exists $k \in n$ and $\mu \in \mathbb{F}_q^*$ such that

$$\iota(\lambda e^{(i)}) = \mu e^{(k)}.$$

This follows from $1 = \mathrm{wt}(\lambda e^{(i)}) = \mathrm{wt}(\iota(\lambda e^{(i)}))$.

3. Moreover, this index k does not depend on λ: Suppose that for $\lambda = 1$ we have $\iota(e^{(i)}) = \nu e^{(j)}$. Then, for $\lambda \neq 1$ we get

$$1 = d(\lambda e^{(i)}, e^{(i)}) = d(\iota(\lambda e^{(i)}), \iota(e^{(i)})) = d(\mu e^{(k)}, \nu e^{(j)}),$$

and this implies $j = k$.

4. Thus we obtain, for the index j defined by $\iota(e^{(i)}) = \nu e^{(j)}$,

$$\iota(\mathbb{F}_q^*(e^{(i)})) \subseteq \mathbb{F}_q^*(e^{(j)}).$$

The bijectivity of ι implies that $\iota(\mathbb{F}_q^*(e^{(i)}))$ is in fact *equal* to $\mathbb{F}_q^*(e^{(j)})$, and it assures the existence of some $\pi \in S_n$ which satisfies

$$\iota\,(\mathbb{F}_q^*(e^{(i)})) = \mathbb{F}_q^*(e^{(\pi(i))}),$$

for all $i \in n$. □

1.5.3 **Lemma** *Let ι be an isometry of $\mathbb{F}_q^n$ with $\iota(0) = 0$. For $i \neq k$ and $\lambda, \mu \in \mathbb{F}_q^*$ we have,*

$$\iota(\lambda e^{(i)} + \mu e^{(k)}) = \iota(\lambda e^{(i)}) + \iota(\mu e^{(k)}).$$

Proof: 1. The assumption implies that

$$2 = \mathrm{wt}(\lambda e^{(i)} + \mu e^{(k)}) = \mathrm{wt}(\iota(\lambda e^{(i)} + \mu e^{(k)})),$$

and so

$$\iota(\lambda e^{(i)} + \mu e^{(k)}) = \nu e^{(j_i)} + \rho e^{(j_k)},$$

for suitable $\nu, \rho \in \mathbb{F}_q^*$ and $j_i \neq j_k$.

2. Using

$$\begin{aligned} 1 &= d(\lambda e^{(i)}, \lambda e^{(i)} + \mu e^{(k)}) = d(\mu e^{(k)}, \lambda e^{(i)} + \mu e^{(k)}) \\ &= d(\iota(\lambda e^{(i)}), \iota(\lambda e^{(i)} + \mu e^{(k)})) = d(\iota(\mu e^{(k)}), \iota(\lambda e^{(i)} + \mu e^{(k)})) \end{aligned}$$

we can deduce from 1. that

$$1 = d(\iota(\lambda e^{(i)}), \nu e^{(j_i)} + \rho e^{(j_k)}) = d(\iota(\mu e^{(k)}), \nu e^{(j_i)} + \rho e^{(j_k)}).$$

Thus, by $j_i \neq j_k$, either $\iota(\lambda e^{(i)}) = \nu e^{(j_i)}$ or $\iota(\lambda e^{(i)}) = \rho e^{(j_k)}$, and similarly either $\iota(\mu e^{(k)}) = \rho e^{(j_k)}$ or $\iota(\mu e^{(k)}) = \nu e^{(j_i)}$.

3. Since ι permutes the orbits of the unit vectors, by 1.5.2, we get from 2. that

$$\iota(\lambda e^{(i)}) + \iota(\mu e^{(k)}) = \nu e^{(j_i)} + \rho e^{(j_k)} = \iota(\lambda e^{(i)} + \mu e^{(k)}),$$

as stated. □

Generalizing this approach we prove

Corollary *Let ι be an isometry of $\mathbb{F}_q^n$ with $\iota(0) = 0$, then, for $v \in \mathbb{F}_q^n$,* **1.5.4**

$$\iota(v) = \iota\left(\sum_{i \in n} v_i e^{(i)}\right) = \sum_{i \in n} \iota(v_i e^{(i)}).$$

Proof: Let k be the number of nonzero components of v. For $0 \leq k \leq 2$ the assertion is true by assumption, by 1.5.2 and 1.5.3. Now we consider $2 < k \leq n$ and assume that the assertion is valid for all vectors with at most $k-1$ nonzero components. We prove that it holds true for the vector $v = \sum_{r \in k} v_{i_r} e^{(i_r)}$ with k nonzero components. Thus we assume that $i_r \in n$ for $r \in k$, $i_r \neq i_s$ for $r, s \in k$, $r \neq s$, and $v_{i_r} \neq 0$ for $r \in k$. Then

$$d\left(\sum_{r=1}^{k-1} v_{i_r} e^{(i_r)}, \sum_{r=0}^{k-1} v_{i_r} e^{(i_r)}\right) = 1 = d\left(\sum_{r=0}^{k-2} v_{i_r} e^{(i_r)}, \sum_{r=0}^{k-1} v_{i_r} e^{(i_r)}\right)$$

whence

$$d\left(\iota\left(\sum_{r=1}^{k-1} v_{i_r} e^{(i_r)}\right), \iota\left(\sum_{r=0}^{k-1} v_{i_r} e^{(i_r)}\right)\right) = 1 = d\left(\iota\left(\sum_{r=0}^{k-2} v_{i_r} e^{(i_r)}\right), \iota\left(\sum_{r=0}^{k-1} v_{i_r} e^{(i_r)}\right)\right)$$

and by the induction hypothesis

$$d\left(\sum_{r=1}^{k-1} \iota\big(v_{i_r} e^{(i_r)}\big), \iota\left(\sum_{r=0}^{k-1} v_{i_r} e^{(i_r)}\right)\right) = 1 = d\left(\sum_{r=0}^{k-2} \iota\big(v_{i_r} e^{(i_r)}\big), \iota\left(\sum_{r=0}^{k-1} v_{i_r} e^{(i_r)}\right)\right).$$

According to 1.5.2 there exists some $\pi \in S_n$ and $\tilde{v}_{i_r} \in \mathbb{F}_q^*$, $r \in k$, so that

$$\sum_{r=1}^{k-1} \iota\big(v_{i_r} e^{(i_r)}\big) = \sum_{r=1}^{k-1} \tilde{v}_{i_r} e^{(\pi(i_r))} \quad \text{and} \quad \sum_{r=0}^{k-2} \iota\big(v_{i_r} e^{(i_r)}\big) = \sum_{r=0}^{k-2} \tilde{v}_{i_r} e^{(\pi(i_r))}.$$

Therefore, necessarily we have

$$\iota\Big(\sum_{r=0}^{k-1} v_{i_r} e^{(i_r)}\Big) = \sum_{r=0}^{k-1} \tilde{v}_{i_r} e^{(\pi(i_r))} = \sum_{r=0}^{k-1} \iota\big(v_{i_r} e^{(i_r)}\big). \qquad \square$$

We are now in a position to describe the group of isometries ι which satisfy $\iota(0) = 0$ as a wreath product. Since

$$\iota(v_i e^{(i)}) \in \iota(\mathbb{F}_q^*(e^{(i)})) = \mathbb{F}_q^*(e^{(\pi(i))}),$$

we can obtain the scalar factor of $e^{(\pi(i))}$ in $\iota(v_i e^{(i)})$ (if $v_i \neq 0$, otherwise we can simply neglect this summand since $\iota(0) = 0$) by the application of a suitable permutation $\varphi(\pi(i))$ of the scalars that keeps 0 fixed,

$$\iota(v_i e^{(i)}) = \varphi(\pi(i))(v_i) e^{(\pi(i))}.$$

Or, in formal terms and since we have to take all the indices into account, there exists a mapping

$$\varphi \colon n \to S_{\mathbb{F}_q^*},$$

from n to the symmetric group

$$S_{\mathbb{F}_q^*} := \{\rho \mid \rho \colon \mathbb{F}_q \to \mathbb{F}_q, \ \rho \text{ is bijective and } \rho(0) = 0\}$$

on $\mathbb{F}_q^*$ (considered as the subgroup of the symmetric group $S_{\mathbb{F}_q}$ on $\mathbb{F}_q$ consisting of the permutations ρ of $\mathbb{F}_q$ that keep the zero element fixed: $\rho(0) = 0$), which satisfies

$$\iota(v_0, \ldots, v_{n-1}) = (\varphi(0)(v_{\pi^{-1}(0)}), \ldots, \varphi(n-1)(v_{\pi^{-1}(n-1)})).$$

This proves the following useful description of the group of isometries:

1.5.5 **Theorem** *The group of isometries ι, with $\iota(0) = 0$, of the finite vector space $\mathbb{F}_q^n$, is the wreath product*

$$S_{\mathbb{F}_q^*} \wr_n S_n$$

of the symmetric group $S_{\mathbb{F}_q^*}$ on $\mathbb{F}_q$ and the symmetric group S_n on n. The action is the following one:

$$S_{\mathbb{F}_q^*} \wr_n S_n \times \mathbb{F}_q^n \to \mathbb{F}_q^n \ : \ ((\varphi;\pi), v) \mapsto (\varphi(0)(v_{\pi^{-1}(0)}), \ldots, \varphi(n-1)(v_{\pi^{-1}(n-1)})).$$

□

It is easy to check that all these $(\varphi;\pi) \in S_{\mathbb{F}_q^*} \wr_n S_n$ are isometries which map 0 onto 0. Together with 1.5.1 we obtain

Theorem *The group of all isometries ι on the finite vector space $\mathbb{F}_q^n$ is the wreath product* 1.5.6

$$S_{\mathbb{F}_q} \wr_n S_n$$

of the symmetric group $S_{\mathbb{F}_q}$ on $\mathbb{F}_q$ and the symmetric group S_n on n. The action is the following one:

$$S_{\mathbb{F}_q} \wr_n S_n \times \mathbb{F}_q^n \to \mathbb{F}_q^n \;:\; ((\varphi;\pi),v) \mapsto (\varphi(0)(v_{\pi^{-1}(0)}),\ldots,\varphi(n-1)(v_{\pi^{-1}(n-1)})).$$

□

It is easy to check that all these $(\varphi;\pi) \in S_{\mathbb{F}_q} \wr_n S_n$ are isometries.

There exist isometries of $\mathbb{F}_q^n$ such that the image of a subspace of $\mathbb{F}_q^n$ is not a subspace. For instance, if $\iota(0) \neq 0$, then the null space $\{0\}$ is not mapped onto a subspace of $\mathbb{F}_q^n$. If $\iota(0) = 0$ consider, for example, the linear $(2,1)$-code C over $\mathbb{F}_5 = \mathbb{Z}/5\mathbb{Z}$ with generator matrix $\Gamma = (1\,1)$. It contains the five codewords $(0,0)$, $(1,1)$, $(2,2)$, $(3,3)$, and $(4,4)$. The image of C under the isometry $\iota = (\varphi;\mathrm{id}) \in S_{\mathbb{F}_q^*} \wr_n S_n$ with $\varphi(0) = \mathrm{id}_{\mathbb{F}_q}$ and $\varphi(1) = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 3 & 2 & 1 & 4 \end{pmatrix}$ is $\{(0,0),(1,3),(2,2),(3,1),(4,4)\}$, which is not a subspace of $\mathbb{F}_5^2$.

Now we want to show that isometries which *map subspaces onto subspaces* belong to the following class of mappings, if $n \geq 3$:

Definition (semilinear mappings) The mapping $\sigma\colon \mathbb{F}_q^n \to \mathbb{F}_q^n$ is called *semilinear* if there exists an automorphism α of $\mathbb{F}_q$ such that, for all $u,v \in \mathbb{F}_q^n$ and all $\kappa \in \mathbb{F}_q$ we have 1.5.7

$$\sigma(u+v) = \sigma(u) + \sigma(v), \qquad \sigma(\kappa u) = \alpha(\kappa)\sigma(u).$$

An isometry which is also a semilinear mapping is called *semilinear isometry* (with respect to the Hamming metric). ◇

Lemma *If the isometry $\iota\colon \mathbb{F}_q^n \to \mathbb{F}_q^n$, $n \geq 3$, maps subspaces onto subspaces, then for each $u \in \mathbb{F}_q^n$ we have* 1.5.8

$$\iota(\mathbb{F}_q^*(u)) = \mathbb{F}_q^*(\iota(u)).$$

Moreover, there exists an automorphism α of $\mathbb{F}_q$ such that, for each $\kappa \in \mathbb{F}_q$,

$$\iota(\kappa u) = \alpha(\kappa)\iota(u).$$

Proof: 1. Since ι maps subspaces onto subspaces, the space $\{0\}$ must be mapped onto itself, whence $\iota(0) = 0$. Therefore, the assertion is obviously true for $u = 0$.

2. Assume that $u \neq 0$. Since ι is bijective and since it maps subspaces to subspaces, $\iota(\langle u\rangle)$ is a one-dimensional subspace, and so, using $\iota(u) \neq 0$, we obtain

$$\iota(\langle u\rangle) = \langle \iota(u)\rangle.$$

Moreover, as $\iota(0) = 0$,

$$\iota(\mathbb{F}_q^*(u)) = \mathbb{F}_q^*(\iota(u)).$$

Hence, there is a permutation of the scalars

$$\Phi_u \in S_{\mathbb{F}_q^*} \leq S_{\mathbb{F}_q},$$

depending possibly on the vector u, which satisfies

$$\iota(\kappa u) = \Phi_u(\kappa)\iota(u).$$

We have to show that Φ_u is independent of u and that it is a field automorphism.

3. For the special case $e := \sum_{i\in n} e^{(i)}$ we have

$$\iota(\kappa e) = \Phi_e(\kappa)\iota(e) = \Phi_e(\kappa)\sum_{i\in n}\varphi(\pi(i))(1)e^{(\pi(i))}, \qquad \kappa \in \mathbb{F}_q^*,$$

as well as

$$\iota(\kappa e) = \sum_{i\in n}\varphi(\pi(i))(\kappa)e^{(\pi(i))}, \qquad \kappa \in \mathbb{F}_q^*,$$

so that we obtain

1.5.9 $$\forall\, i \in n \;:\; \Phi_e(\kappa) = \frac{\varphi(\pi(i))(\kappa)}{\varphi(\pi(i))(1)}, \qquad \kappa \in \mathbb{F}_q^*.$$

4. Now we prove that $\Phi_e(\kappa\mu) = \Phi_e(\kappa)\Phi_e(\mu)$, for $\kappa, \mu \in \mathbb{F}_q$. The assertion is trivial for $\kappa = 0$ or $\mu = 0$. So it is possible to restrict attention to $\kappa, \mu \in \mathbb{F}_q^*$. To begin with, we consider another special case (recalling that $n > 2$, by assumption): Let

$$w := e^{(0)} + \mu e^{(i)},$$

for $i \neq 0$ and $\mu \in \mathbb{F}_q^*$. The corresponding equation

$$\iota(\kappa w) = \Phi_w(\kappa)\iota(w), \qquad \kappa \in \mathbb{F}_q^*,$$

implies that

$$\begin{aligned}\varphi(\pi(0))(\kappa)e^{(\pi(0))} &+ \varphi(\pi(i))(\kappa\mu)e^{(\pi(i))}\\ &= \Phi_w(\kappa)\big(\varphi(\pi(0))(1)e^{(\pi(0))} + \varphi(\pi(i))(\mu)e^{(\pi(i))}\big).\end{aligned}$$

Comparing the coefficients of the basis vectors on both sides we obtain two useful identities. The coefficients of $e^{(\pi(0))}$ give

$$\varphi(\pi(0))(\kappa) = \Phi_w(\kappa)\varphi(\pi(0))(1),$$

so that we can deduce

$$\Phi_w(\kappa) = \frac{\varphi(\pi(0))(\kappa)}{\varphi(\pi(0))(1)} = \Phi_e(\kappa), \qquad \kappa \in \mathbb{F}_q^*,$$

and hence $\Phi_w = \Phi_e$ in this particular situation. The second identity, obtained by comparing the coefficients of $e^{(\pi(i))}$, is

$$\varphi(\pi(i))(\kappa\mu) = \Phi_w(\kappa)\varphi(\pi(i))(\mu).$$

Using $\Phi_w = \Phi_e$ and dividing both sides by $\varphi(\pi(i))(1)$ we derive that

$$\Phi_e(\kappa\mu) = \Phi_e(\kappa)\Phi_e(\mu), \qquad \kappa, \mu \in \mathbb{F}_q^*,$$

i.e. Φ_e is multiplicative.

5. We want to show that $\Phi_u = \Phi_e$, for all $u \neq 0$. According to 1.5.4 and 1.5.9, for $u = \sum_{i\in n} u_i e^{(i)}$ we get

$$\begin{aligned} \iota(u) &= \sum_{i\in n} \iota(u_i e^{(i)}) = \sum_{i\in n} \varphi(\pi(i))(u_i) e^{(\pi(i))} \\ &= \sum_{i\in n} \Phi_e(u_i)\varphi(\pi(i))(1) e^{(\pi(i))}. \end{aligned}$$

Since Φ_e is multiplicative, we derive for $\kappa \in \mathbb{F}_q^*$ that

$$\begin{aligned} \iota(\kappa u) &= \sum_{i\in n} \Phi_e(\kappa u_i)\varphi(\pi(i))(1) e^{(\pi(i))} \\ &= \Phi_e(\kappa)\sum_{i\in n} \Phi_e(u_i)\varphi(\pi(i))(1) e^{(\pi(i))} \\ &= \Phi_e(\kappa)\iota(u), \end{aligned}$$

which can be compared with the identity

$$\iota(\kappa u) = \Phi_u(\kappa)\iota(u),$$

obtaining $\Phi_e(\kappa) = \Phi_u(\kappa)$ for all $\kappa \in \mathbb{F}_q^*$. Hence we have proved that in fact $\Phi_u = \Phi_e$, as stated.

6. It remains to show that Φ_e is additive, i.e.

$$\Phi_e(\lambda + \mu) = \Phi_e(\lambda) + \Phi_e(\mu), \qquad \lambda, \mu \in \mathbb{F}_q.$$

Since $\Phi_e(0) = 0$, this formula is true for $\lambda = 0$ or $\mu = 0$. By assumption $n \geq 3$, and so we can consider

$$u := e^{(0)} + e^{(1)}, \;\; w := e^{(1)} + e^{(2)}$$

and the subspace $U := \langle\{u,w\}\rangle$ generated by these two vectors. For $\lambda,\mu \in \mathbb{F}_q^*$, the vectors $\iota(\lambda u)$, $\iota(\mu w)$ and $\iota(\lambda u)+\iota(\mu w)$ are contained in the subspace $\iota(U)$. Hence, there exists some $z \in U$, for which $\iota(z) = \iota(\lambda u)+\iota(\mu w)$. Then

$$\begin{aligned}\iota(z) \quad &= \quad \Phi_e(\lambda)\varphi(\pi(0))(1)e^{(\pi(0))} + \Phi_e(\lambda)\varphi(\pi(1))(1)e^{(\pi(1))} \\ &\quad + \Phi_e(\mu)\varphi(\pi(1))(1)e^{(\pi(1))} + \Phi_e(\mu)\varphi(\pi(2))(1)e^{(\pi(2))}.\end{aligned}$$

On the other hand, since

$$z = z_0e^{(0)} + (z_0+z_2)e^{(1)} + z_2e^{(2)},$$

we have

$$\begin{aligned}\iota(z) \quad &= \quad \Phi_e(z_0)\varphi(\pi(0))(1)e^{(\pi(0))} + \Phi_e(z_0+z_2)\varphi(\pi(1))(1)e^{(\pi(1))} \\ &\quad + \Phi_e(z_2)\varphi(\pi(2))(1)e^{(\pi(2))}.\end{aligned}$$

Since $\varphi(\pi(i))(1) \neq 0$, we derive from these two representations of $\iota(z)$ that $\Phi_e(z_0) = \Phi_e(\lambda)$ and $\Phi_e(z_2) = \Phi_e(\mu)$. Since Φ_e is a bijection on $\mathbb{F}_q$, we obtain $z_0 = \lambda$, $z_2 = \mu$ and

$$\Phi_e(\lambda) + \Phi_e(\mu) = \Phi_e(z_0+z_2) = \Phi_e(\lambda+\mu),$$

which completes the proof of the additivity.

7. Hence, $\alpha := \Phi_e$ is in fact an automorphism of $\mathbb{F}_q$ which satisfies

$$\iota(\kappa u) = \alpha(\kappa)\iota(u), \qquad \kappa \in \mathbb{F}_q,\ u \in \mathbb{F}_q^n.$$

Finally

$$\begin{aligned}\iota(u+v) &= \iota\Big(\sum_{i\in n}(u_i+v_i)e^{(i)}\Big) \\ &= \sum_{i\in n}\iota\big((u_i+v_i)e^{(i)}\big) \\ &= \sum_{i\in n}\alpha(u_i+v_i)\varphi(\pi(i))(1)e^{(\pi(i))} \\ &= \sum_{i\in n}\alpha(u_i)\varphi(\pi(i))(1)e^{(\pi(i))} + \sum_{i\in n}\alpha(v_i)\varphi(\pi(i))(1)e^{(\pi(i))} \\ &= \iota(u)+\iota(v),\end{aligned}$$

which completes the proof. □

Summarizing, an isometry of $\mathbb{F}_q^n$, $n \geq 3$, which maps subspaces onto subspaces is semilinear and is described by three mappings

$$\varphi\colon n \to S_{\mathbb{F}_q^*},\ \ \alpha \in \operatorname{Aut}(\mathbb{F}_q),\ \ \pi \in S_n.$$

It acts on a vector $v \in \mathbb{F}_q^n$ by

$$\iota(v_0,\ldots,v_{n-1}) = (\alpha(v_{\pi^{-1}(0)})\varphi(0)(1),\ldots,\alpha(v_{\pi^{-1}(n-1)})\varphi(n-1)(1)).$$

The permutations $\varphi(i)$ are contained in $S_{\mathbb{F}_q^*}$, and so each factor $\varphi(i)(1)$ is contained in $\mathbb{F}_q^*$. Since we only need to know the values $\varphi(i)(1)$, $i \in n$, we can replace the mapping φ by

$$\psi\colon n \to \mathbb{F}_q^* \;:\; \psi(i) := \varphi(i)(1), \qquad i \in n.$$

Therefore, we can write ι as the triple $(\psi;(\alpha,\pi))$, where $(\psi;\pi)$ is a linear isometry. In other words $(\psi;\pi)$ belongs to the wreath product $\mathbb{F}_q^* \wr_n S_n$. This allows the slightly simpler expression for $\iota(v)$ given by

$$(\psi;(\alpha,\pi))(v_0,\ldots,v_{n-1}) = (\alpha(v_{\pi^{-1}(0)})\psi(0),\ldots,\alpha(v_{\pi^{-1}(n-1)})\psi(n-1)).$$

We collect these results in the following

Theorem *For $n \geq 3$, the isometries of $\mathbb{F}_q^n$ which map subspaces onto subspaces are exactly the semilinear mappings of the form $(\psi;(\alpha,\pi))$, where $(\psi;\pi)$ is a linear isometry and α is a field automorphism. These mappings form a group, the* group of semilinear isometries. □ **1.5.10**

In Section 6.7, we will describe this group as a generalized wreath product.

Definition (semilinearly isometric codes) Two (n,k)-codes C and C' over $\mathbb{F}_q$ are called *semilinearly isometric* if and only if there exists an automorphism α in $\mathrm{Aut}(\mathbb{F}_q)$ and a linear isometry $(\psi;\pi)$ in $\mathbb{F}_q^* \wr_n S_n$, such that the mapping **1.5.11**

$$(c_0,\ldots,c_{n-1}) \mapsto \big(\psi(0)\alpha(c_{\pi^{-1}(0)}),\ldots,\psi(n-1)\alpha(c_{\pi^{-1}(n-1)})\big)$$

maps C onto C'. The orbits of the group of semilinear isometries on the set of subspaces of $H(n,q)$ are the *semilinear isometry classes* of linear codes of length n over $\mathbb{F}_q$. ◇

In addition, we mention the following facts (the first one is obvious, the second one will become clear in the chapter on finite fields):

1. The group of linear isometries of $H(n,2)$ is isomorphic to the symmetric group S_n, since $\mathbb{F}_2^* = \{1\}$.
2. The group of semilinear isometries of $H(n,q)$ is the same as the group of linear isometries if and only if q is a prime p. The reason is that the field $\mathbb{F}_q$ has only the trivial automorphism if and only if $q = p$.

Hence, if the linear and semilinear isometry groups differ, we expect to see different numbers of orbits. This is indeed the case. The smallest examples are for $q = 4$, $n = 8$ and $k \geq 3$ (see Tables 6.9 and 6.31 in Chapter 6). What happens is that two linear isometry classes form a single semilinear isometry class. For example, we consider two codes over a field consisting of four elements. We take the field $\mathbb{F}_4 = \{0, 1, \alpha, \alpha + 1\}$ subject to the relation $\alpha^2 = \alpha + 1$ (see Chapter 3 for more details on finite fields). The code C_1 generated by

$$\Gamma_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \alpha & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ \alpha & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \alpha+1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

is semilinearly equivalent to C_2 generated by

$$\Gamma_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \alpha & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ \alpha & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \alpha & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

To see that the codes are semilinearly equivalent, add the first row of Γ_1 to the second and third row. This gives

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

Now swap pairwise the second and the fifth and the third and the fourth column to get

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

Application of the field automorphism $x \mapsto x^2$ takes the resulting matrix to Γ_2. It can be proved that the codes C_1 and C_2 are linearly inequivalent. This shows that two linear isometry classes may be combined under the semilinear group. Indeed, the number of linear isometry classes of codes which may join is bounded from above by the number of field automorphisms, which is of course two in this case.

For $n = 1$ or $n = 2$ the groups of isometries that map subspaces onto subspaces are described in

1.5.12 **Theorem** *For $n = 1$, the isometries of $\mathbb{F}_q$ which map subspaces onto subspaces are exactly the isometries of $\mathbb{F}_q$ which map 0 onto 0. According to 1.5.5 these are the elements of $S_{\mathbb{F}_q^*}$.*

For $n = 2$, the isometries of $\mathbb{F}_q^2$ which map subspaces onto subspaces are exactly the mappings of the form $(\psi; (\alpha, \pi))$, where $(\psi; \pi)$ is a linear isometry and α is a group automorphism of the multiplicative group $\mathbb{F}_q^*$. □

One can check that there exist group automorphisms of $\mathbb{F}_q^*$ which cannot be extended to field automorphisms of $\mathbb{F}_q$.

In conclusion, in large parts of the present book we will be concerned with orbits of the linear or semilinear isometry group on the set of subspaces of $\mathbb{F}_q^n = H(n, q)$.

Exercises

Exercise Prove 1.5.1. **E.1.5.1**

Exercise Complete the proofs of 1.5.5 and 1.5.6 by showing that all elements of the corresponding wreath products are isometries. **E.1.5.2**

Exercise In order to complete the proof of 1.5.10, show that any semilinear mapping of the given form is an isometry which maps subspaces onto subspaces. Moreover, prove 1.5.12. **E.1.5.3**

1.6 The Weight Enumerator

1.6

An important issue is to find out when two $k \times n$ generator matrices over $\mathbb{F}_q$ define isometric codes. In general, this is not an easy task since normal forms of generator matrices are expensive to find (cf. Chapter 9). But there are *invariants* of linear codes which may help to distinguish between different codes. An invariant is simply a quantity (or a property) which we can associate to a code, and which is equal for codes of the same equivalence class (i.e. a "fingerprint"). One of these invariants will be introduced next, it is the weight distribution of a code. Essentially, this distribution records how many words of a code have a given Hamming weight. It is usually recorded as the coefficients of a polynomial, the weight enumerator. The permutational, linear or semilinear isometries of 1.4 and 1.5 preserve Hamming distances and Hamming weights. Codes with different weight enumerators are definitely not permutationally, linearly or semilinearly isometric. In Chapter 8, we will introduce a method for the evaluation of generator matrices that automatically provides the weight distribution as well.

We display the weight distribution of a linear code C of length n in terms of a *generating polynomial*. For this purpose we use commuting indeterminates

x and y over $\mathbb{C}$, and we indicate by $A_i = A_i(C)$ the number of codewords of weight i in C. For example, if $\operatorname{dist}(C) = d > 1$, then $A_0 = 1$, $A_1 = \ldots = A_{d-1} = 0$ and $A_d \neq 0$.

1.6.1 **Definition (weight enumerator)** The homogeneous *weight enumerator* of a linear code C of length n is defined as

$$W_C(x,y) := \sum_{c \in C} x^{\mathrm{wt}(c)} y^{n-\mathrm{wt}(c)} = \sum_{i=0}^{n} A_i x^i y^{n-i} \in \mathbb{C}\,[x,y].$$

Notice that this is a homogeneous polynomial of degree n. Setting $y = 1$ yields the inhomogeneous weight enumerator

$$w_C(x) := \sum_{c \in C} x^{\mathrm{wt}(c)} = \sum_{i=0}^{n} A_i x^i \in \mathbb{C}\,[x]. \qquad \diamond$$

For example, the 4-fold binary repetition code

$$C = \{\mathbf{0}_4, \mathbf{1}_4\}$$

has weight enumerator

$$W_C(x,y) = x^4 + y^4 \text{ and } w_C(x) = x^4 + 1.$$

The following result is often useful. It follows from Exercise 1.2.14.

1.6.2 **Lemma** *For any two vectors $u, v \in \mathbb{F}_2^n$ we have the equivalence*

$$\mathrm{wt}(u+v) \equiv \mathrm{wt}(v) \bmod 2 \iff \mathrm{wt}(u) \equiv 0 \bmod 2. \qquad \square$$

This means that adding a vector $u \in \mathbb{F}_2^n$ to a vector $v \in \mathbb{F}_2^n$ keeps the congruence class modulo two of $\mathrm{wt}(v)$ fix if and only if the weight of u is even. Hence, adding a vector of odd weight in a binary code C to vectors of even weight gives vectors of odd weight and vice versa. Since the set of vectors of a linear code is closed under addition, this leaves only two possible cases. Either there is no vector of odd weight in a binary code C, or the set of vectors of C falls into two categories of equal size, one consisting of the vectors of even weight and the other containing all vectors whose weight is odd.

1.6.3 **Corollary** *For binary codes C the following holds true:*

- *The codewords of even weight form the subspace*

$$C_e := \{c \in C \mid \mathrm{wt}(c) \text{ is even}\}.$$

- *If there exists a codeword c of odd weight, then the complement $C \setminus C_e$ of C_e is equal to $c + C_e$.*
- *Hence either there is no element of odd weight or exactly half of the codewords in C have odd weight. We can express this fact as follows: If*

$$R := \sum_{i=0}^{\lfloor n-1/2 \rfloor} A_{2i+1} \text{ and } S := \sum_{i=0}^{\lfloor n/2 \rfloor} A_{2i},$$

then either $R = 0$ or $R = S$. □

In 1.6.9 we will derive an identity which is due to MacWilliams. It shows that the weight enumerator of a code and that of its dual code mutually determine each other. In order to prepare for a proof of this identity, we introduce the notion of a *linear* representation of a group. This notion generalizes the concept of a *permutation* representation or action of a group which has already been used on several occasions.

According to 1.4.5, a finite action ${}_GX$ is essentially the same as a *permutation representation* of G on X. This is a homomorphism

$$\delta \colon G \to S_X \; : \; g \mapsto \delta(g),$$

from G into S_X, where $g \in G$ is mapped onto $\delta(g) = \overline{g}$, the permutation $x \mapsto gx$ of X, an element of the symmetric group S_X. A *linear representation* D of G over a field $\mathbb{F}$ is defined to be a homomorphism

$$D \colon G \to \mathrm{GL}(V) \; : \; g \mapsto D(g),$$

from G into the group $\mathrm{GL}(V)$ of invertible linear mappings on a finite dimensional vector space V over $\mathbb{F}$. The vector space V is called the *representation space* and its dimension f^D is called the *dimension* of D. $\mathbb{F}$ is said to be the *groundfield* of D.

Two representations $D \colon G \to \mathrm{GL}(V)$ and $D' \colon G \to \mathrm{GL}(V')$ of G over $\mathbb{F}$ are considered *equivalent* if there exists an invertible linear mapping $T \colon V \to V'$ such that

$$\forall\, g \in G \; : \; TD(g) = D'(g)T.$$

Every choice of a basis $\{b^{(0)}, \ldots, b^{(f^D-1)}\}$ of V yields invertible matrices $\mathbf{D}(g)$ which describe $D(g)$ with respect to the given basis. Therefore, a *matrix representation* $\mathbf{D}$ of G over $\mathbb{F}$ is a homomorphism

$$\mathbf{D} \colon G \to \mathrm{GL}_{f^D}(\mathbb{F}) \; : \; g \mapsto \mathbf{D}(g)$$

from G to the *general linear group* $\mathrm{GL}_{f^D}(\mathbb{F})$, the group consisting of all invertible matrices over $\mathbb{F}$ with f^D rows and columns. Conversely, it is clear that each matrix representation $\mathbf{D}\colon G \to \mathrm{GL}_{f^D}(\mathbb{F})$ yields a representation $D\colon G \to \mathrm{GL}(V)$ where V is an f^D-dimensional vector space over $\mathbb{F}$. Equivalence of matrix representations is defined correspondingly. Hence we are free to consider either representations or matrix representations. Which concept we choose will depend on the situation in question. In the present section we are mainly concerned with matrix representations and their characters.

Let D be a representation of G. Consider the map

$$\chi^D\colon G \to \mathbb{F} \ : \ g \mapsto \sum_{i \in f^D} d_{ii}(g) = \mathrm{trace}(\mathbf{D}(g)),$$

which takes $g \in G$ to $\chi^D(g)$, the trace of $\mathbf{D}(g) = (d_{ij}(g))$. From Linear Algebra it is clear that the trace of a matrix $\mathbf{D}(g)$ corresponding to the linear mapping $D(g)$ is independent of the choice of a basis. The map χ^D is called the *character* of D. Representations and characters over the field $\mathbb{C}$ of complex numbers are called *ordinary*. In the case when the groundfield is finite, they are called *modular.*

1.6.4 Examples

- Every finite action ${}_GX$ yields a representation on the space $\mathbb{F}^X$, the vector space over $\mathbb{F}$ which has a basis whose elements are indexed by the elements of X. Thus we already have a wealth of examples at hand.
- The trivial representation of G arising from the *trivial action* ${}_G\{x\}$ of G on a set of cardinality one, where $gx := x$, is called the *identity representation* or the *trivial representation* and it is indicated as

  $$I\colon G \to \mathrm{GL}(\mathbb{F}) \ : \ g \mapsto id_{\mathbb{F}},$$

  where $\mathbb{F}$ is the 1-dimensional vector space over $\mathbb{F}$. Its character χ^I has the value $\chi^I(g) = 1_{\mathbb{F}}$, for each $g \in G$.
- In general, any finite action ${}_GX$ gives rise to a linear representation of G on $\mathbb{F}^X$. This representation associates to g the permutation $\overline{g}$ of the basis elements X (recall 1.4.5). Its character is

  $$\chi(g) = a_1(\overline{g}) := |\{x \in X \mid gx = x\}|, \qquad g \in G,$$

  which counts the number of fixed points of g. More precisely, $\chi(g) = a_1(\overline{g}) \cdot 1_{\mathbb{F}}$. In the *ordinary* case, i.e. if $\mathbb{F} = \mathbb{C}$, this character is the character of the action of the group.
- A *one-dimensional character* of G is the character of a one-dimensional linear representation, whence a homomorphism from G into the multiplicative group $\mathbb{F}^*$ of the groundfield. Therefore, for each such character χ we have

  $$\chi(g \cdot h) = \chi(g)\chi(h) \ \text{ and } \ \chi(1_G) = 1_{\mathbb{F}},$$

provided that G is written multiplicatively. If G is written additively, we have correspondingly

$$\chi(g+h) = \chi(g)\chi(h) \text{ and } \chi(0_G) = 1_{\mathbb{F}}.$$

- For our purposes, the one-dimensional characters are of particular interest. A simple example is a one-dimensional character of the additive group of the field $\mathbb{F}_q$. Consider the group

  $$G := \mathbb{Z}_p := \{\overline{z} \mid z \in p\}$$

  of residue classes $\overline{z}$ of integers $z \in \mathbb{Z}$ modulo the prime number p. For more details on residue classes see Exercise 3.1.3. Addition in the group is done modulo the prime p. If

  $$\xi := e^{\frac{2\pi i}{p}} \in \mathbb{C}$$

  denotes a primitive p-th root of unity, and if $j \in p$, the mapping

  $$\chi^{(j)} : \mathbb{Z}_p \to \mathbb{C}^* \ : \ \overline{z} \mapsto \xi^{j \cdot z}$$

  is a one-dimensional character of G. It is not difficult to see that these characters are in fact *all* one-dimensional characters over $\mathbb{C}$ of this group, but we do not need this fact. We just remark that the character $\chi^{(j)}$ is nontrivial for $j \neq 0$.

- We can easily generalize this to a direct product

  $$G := \mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$$

  of $m \geq 2$ factors of such groups. If $(\overline{z}_0, \ldots, \overline{z}_{m-1}) \in G$, and $j_i \in p$, then

  $$\chi^{(j_0,\ldots,j_{m-1})} : (\overline{z}_0, \ldots, \overline{z}_{m-1}) \mapsto \xi^{\sum_i j_i z_i}$$

  is a one-dimensional character of $G = \mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$. Moreover, this character is nontrivial if and only if $j_i \neq 0$ for at least one i.

- Later on in 3.1.6 we will see that for $q = p^m$ with p prime, the additive group of $\mathbb{F}_q$ is isomorphic to $G := \mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$ (with m factors $\mathbb{Z}_p$). Hence, we have established the existence of nontrivial one-dimensional characters of the additive group of any finite field. This fact is all we need in the present section. ◇

In particular we use the following result on the sum of character values:

Lemma *Let χ be a nontrivial character of a finite group G over a field $\mathbb{F}$. Then* **1.6.5**

$$\sum_{g \in G} \chi(g) = 0.$$

Proof: Since χ is nontrivial, there exists an element $h \in G$ such that $\chi(h) \neq 1$. From

$$\chi(h) \sum_{g \in G} \chi(g) = \sum_{g \in G} \chi(h \cdot g) = \sum_{g \in G} \chi(g),$$

we obtain that

$$\big(\chi(h) - 1\big) \sum_{g \in G} \chi(g) = 0,$$

and this implies the statement since $\chi(h) \neq 1$. □

Suppose that $\chi \colon \mathbb{F}_q \to \mathbb{C}^*$ is a nontrivial one-dimensional ordinary character of $\mathbb{F}_q$, whose existence was established in 1.6.4. Fix an element $0 \neq v \in \mathbb{F}_q^n$. Using the standard bilinear form on $\mathbb{F}_q^n$, we introduce a character of the additive group $G := \mathbb{F}_q^n$ as follows:

1.6.6 $$\chi_{(v)} \colon \mathbb{F}_q^n \to \mathbb{C}^* \ : \ w \mapsto \chi(\langle v, w \rangle).$$

It is not difficult to see that this is a nontrivial one-dimensional character of $\mathbb{F}_q^n$.

Let us return to the weight enumerator W_C and consider the *weight function* in its homogeneous form,

$$f \colon \mathbb{F}_q^n \to \mathbb{C}[x, y] \ : \ v \mapsto x^{\mathrm{wt}(v)} y^{n - \mathrm{wt}(v)}.$$

Together with the weight function we examine a second function, a *Discrete Fourier Transform* of f (see also Exercise 1.6.9). It is defined by

$$\hat{f} := \sum_{v \in \mathbb{F}_q^n} f(v) \cdot \chi_{(v)},$$

where $\chi_{(v)}$ is the character defined by 1.6.6. To begin with, we prove

1.6.7 **Lemma** *For $w \in \mathbb{F}_q^n$ we have*

$$\hat{f}(w) = (y - x)^{\mathrm{wt}(w)} \big(y + (q - 1)x\big)^{n - \mathrm{wt}(w)}.$$

Proof: Let χ denote a nontrivial one-dimensional ordinary character of the additive group $G := \mathbb{F}_q$. For $\alpha \in \mathbb{F}_q$ we define

$$|\alpha| := \begin{cases} 1 & \text{if } \alpha \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

For each $w \in \mathbb{F}_q^n$ we compute

$$\begin{aligned} \hat{f}(w) &= \sum_{v \in \mathbb{F}_q^n} \chi(\langle v, w \rangle) f(v) \\ &= \sum_{v \in \mathbb{F}_q^n} \chi(\langle v, w \rangle) x^{\mathrm{wt}(v)} y^{n - \mathrm{wt}(v)} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{v_0\in\mathbb{F}_q} \cdots \sum_{v_{n-1}\in\mathbb{F}_q} \chi\left(\sum_{i\in n} v_i w_i\right) x^{|v_0|+\ldots+|v_{n-1}|} y^{(1-|v_0|)+\ldots+(1-|v_{n-1}|)} \\
&= \sum_{v_0\in\mathbb{F}_q} \cdots \sum_{v_{n-1}\in\mathbb{F}_q} \prod_{i\in n} \chi(v_i w_i) x^{|v_i|} y^{1-|v_i|} \\
&= \prod_{i\in n} \sum_{g\in G} \chi(g w_i) x^{|g|} y^{1-|g|}.
\end{aligned}
$$

For the fourth equation we used that χ is a homomorphism.

- If $w_i = 0$ then $\chi(gw_i) = \chi(0) = 1$, and so

$$\sum_{g\in G} \chi(gw_i) x^{|g|} y^{1-|g|} = y + (q-1)x.$$

- On the other hand, if $w_i \neq 0$, we obtain

$$
\begin{aligned}
\sum_{g\in G} \chi(gw_i) x^{|g|} y^{1-|g|} &= y + \sum_{g\in G\setminus\{0\}} \chi(gw_i) x \\
&= y + \sum_{g\in G\setminus\{0\}} \chi(g) x
\end{aligned}
$$

which, by 1.6.5, equals $y - \chi(0)x = y - x$. □

Lemma *If C is an (n,k)-code over $\mathbb{F}_q$, then* **1.6.8**

$$\sum_{c\in C} \hat{f}(c) = q^k \sum_{v\in C^\perp} f(v).$$

Proof: We know that

$$
\begin{aligned}
\sum_{c\in C} \hat{f}(c) &= \sum_{c\in C} \sum_{v\in\mathbb{F}_q^n} \chi_{(v)}(c) f(v) \\
&= \sum_{v\in\mathbb{F}_q^n} \sum_{c\in C} \chi(\langle v,c\rangle) f(v) \\
&= \sum_{v\in C^\perp} \sum_{c\in C} \chi(\langle v,c\rangle) f(v) + \sum_{v\in\mathbb{F}_q^n\setminus C^\perp} \sum_{c\in C} \chi(\langle v,c\rangle) f(v).
\end{aligned}
$$

In the first sum we have $\chi(\langle v,c\rangle) = \chi(0) = 1$ for all $v \in C^\perp$ and all $c \in C$. In order to simplify the second sum we recall that the map $c \mapsto \langle v,c\rangle$ is a linear form $C \to \mathbb{F}_q$. Since v belongs to $\mathbb{F}_q^n \setminus C^\perp$, this linear form is surjective, whence its kernel has dimension $k-1$. Therefore, for each $g \in \mathbb{F}_q$, there are q^{k-1} vectors $c \in C$ such that $\langle v,c\rangle = g$. For this reason we can continue as follows:

$$\sum_{c\in C} \hat{f}(c) = q^k \sum_{v\in C^\perp} f(v) + q^{k-1} \sum_{v\in\mathbb{F}_q^n\setminus C^\perp} f(v) \sum_{g\in G} \chi(g) = q^k \sum_{v\in C^\perp} f(v),$$

by 1.6.5. □

We are now in a position to prove the announced identity of MacWilliams [137] for the weight distribution of the dual code:

1.6.9 **The MacWilliams-identity** *The weight enumerator of an (n,k)-code C over $\mathbb{F}_q$ is related to the weight enumerator of its dual code in the following way:*

$$W_{C^\perp}(x,y) = q^{-k} W_C(y-x, y+(q-1)x).$$

Proof:

$$W_{C^\perp}(x,y) = \sum_{c\in C^\perp} f(c) \overset{1.6.8}{=} q^{-k} \sum_{c\in C} \hat{f}(c) \overset{1.6.7}{=} q^{-k} W_C(y-x, y+(q-1)x). \qquad \Box$$

1.6.10 **Example** Recall from 1.6.1 that the 4-fold binary repetition code $C = \{\mathbf{0}_4, \mathbf{1}_4\}$ has weight enumerator $W_C(x,y) = x^4 + y^4$. By the MacWilliams-identity, the weight enumerator of its dual code is

$$W_{C^\perp}(x,y) = \frac{1}{2}\big((y-x)^4 + (y+x)^4\big) = y^4 + 6x^2y^2 + x^4. \qquad \diamond$$

It is sometimes useful to apply the MacWilliams-identity with exchanged roles of C and $C^\perp$.

1.6.11 **Example** Consider the $(7,4)$-Hamming-code of 1.3.6. The dual code $C^\perp$, generated by Δ, has 8 codewords. The 7 nonzero words are all of weight 4. Hence $C^\perp$ has weight enumerator

$$W_{C^\perp}(x,y) = y^7 + 7x^4y^3.$$

By the MacWilliams-identity 1.6.9, the weight enumerator of the $(7,4)$ Hamming-code $C^{\perp\perp} = C$ is determined as

$$\begin{aligned} W_C(x,y) &= \frac{1}{2^3} W_{C^\perp}(y-x, y+x) \\ &= \frac{1}{8}\big((y+x)^7 + 7(y-x)^4(y+x)^3\big) \\ &= y^7 + 7x^3y^4 + 7x^4y^3 + y^7. \end{aligned}$$

This shows that C has 7 words of weight 3 and 4 each. Together with the zero and the all-one-vector, this amounts to all 16 words in the code. $\diamond$

Particular cases of interest are the *self-dual codes* which we have introduced in 1.3.3. These are the linear codes C satisfying $C = C^\perp$. For these codes we have $k = n-k$, n is therefore even and $k = n/2$. Since the weight enumerator is a homogeneous polynomial of degree n, this implies the following:

Corollary *If C is self-dual, then* **1.6.12**

$$W_C(x,y) = W_C(-x,-y) = W_C\left(\frac{y-x}{\sqrt{q}}, \frac{y+(q-1)x}{\sqrt{q}}\right). \qquad \square$$

This corollary shows that the weight enumerator of a self-dual code is a fixed point, i.e. an invariant of a group acting on a polynomial ring, in the following sense:

Definition (fixed point, invariant) Let ${}_GX$ be an action of a group G on a set X. An element $x \in X$ is called a *fixed point* of an element $g \in G$ if $gx = x$. The set of *all fixed points of* g is denoted by **1.6.13**

$$X_g := \{x \in X \mid gx = x\},$$

and we let

$$X_G := \left\{x \in X \;\middle|\; \forall\, g \in G \;:\; gx = x\right\} = \bigcap_{g \in G} X_g$$

be the set of common fixed points of all elements $g \in G$. The elements in X_G are also called the *invariants* of G on X. ◇

We note that the linear group $\mathrm{GL}_n(\mathbb{C})$ of invertible matrices of rank n over the complex field acts on the polynomial ring $\mathbb{C}[x_0, \ldots, x_{n-1}]$ in the following way:

$$\mathrm{GL}_n(\mathbb{C}) \times \mathbb{C}[x_0, \ldots, x_{n-1}] \to \mathbb{C}[x_0, \ldots, x_{n-1}],$$
$$(B, f(x_0, \ldots, x_{n-1})) \mapsto (Bf)(x_0, \ldots, x_{n-1}) := f((x_0, \ldots, x_{n-1}) \cdot B^\top).$$

For example, if $B := -I_2 := \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ then

$$\big((-I_2)W_C\big)(x,y) = W_C\Big((x,y) \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}\Big) = W_C(-x,-y),$$

which shows that the weight enumerator W_C of a self-dual code is a fixed point of $-I_2 \in \mathrm{GL}_2(\mathbb{C})$. We may also express this by saying that W_C is an invariant of the group $G := \{I_2, -I_2\}$ of order two.

Any subgroup $G \leq \mathrm{GL}_n(\mathbb{C})$ induces a subaction, and the set of common fixed points

$$\mathbb{C}[x_0, \ldots, x_{n-1}]_G = \left\{ f \in \mathbb{C}[x_0, \ldots, x_{n-1}] \;\middle|\; \forall\, B \in G : Bf = f \right\}$$

is the set of *invariants* of G on $\mathbb{C}[x_0, \ldots, x_{n-1}]$. The standard example is

$$\mathbb{C}[x_0, \ldots, x_{n-1}]_{S_n},$$

the set of invariants of the symmetric group S_n. A polynomial in this set is invariant under all possible permutations of its variables. Hence the invariants of the symmetric group consist of the *symmetric polynomials.*

An important series of symmetric polynomials is the series of *elementary symmetric* polynomials

$$\sigma_m := \sum_{0\le i_1<\ldots<i_m\le n-1} x_{i_1}\cdots x_{i_m}, \quad 1\le m\le n,\ \sigma_0 := 1.$$

They generate the ring of symmetric polynomials, i.e. any symmetric polynomial can be written in a unique way as a polynomial in the elementary symmetric polynomials. Moreover, the coefficients of polynomials can be expressed in terms of their roots, using elementary symmetric polynomials. For example

$$\prod_{i\in n}(x-\kappa_i) = \sigma_0\cdot x^n + \ldots + (-1)^n\sigma_n(\kappa_0,\ldots,\kappa_{n-1}).$$

For the other coefficients see Exercise 1.6.13.

From 1.6.12 we derive that the weight enumerator W_C of a self-dual linear code C is an invariant of the group

$$G := \langle -I_2,\ B\rangle \quad \text{where} \quad B = \frac{1}{\sqrt{q}}\begin{pmatrix} -1 & 1 \\ q-1 & 1 \end{pmatrix}.$$

It is easy to check that this group has four elements. It is isomorphic to the Klein four-group V_4, and so we have obtained:

1.6.14 **Corollary** *The weight enumerator of a self-dual linear code is an invariant of the Klein four-group:*

$$W_C(x,y)\in\mathbb{C}\,[x,y]_{V_4}. \qquad \square$$

Binary self-dual codes have an additional property:

1.6.15 **Definition (divisible codes)** A linear code C is called *r-divisible* if each codeword has a weight which is divisible by r. ◇

A 2-divisible code is called *even*, a 4-divisible code is called *doubly even.* A code which is 2-divisible but not 4-divisible is called *singly even.* Notice that a binary self-dual code is even, since each word is orthogonal to itself, which means that its Hamming weight is even.

1.6.16 **Lemma** *If C is self-dual and r-divisible, then its weight distribution W_C is an invariant of the group*

$$G := \langle -I_2, B, D\rangle\,, \ \textit{where} \ B = \frac{1}{\sqrt{q}}\begin{pmatrix} -1 & 1 \\ q-1 & 1 \end{pmatrix},\ D = \begin{pmatrix} \epsilon & 0 \\ 0 & 1 \end{pmatrix}$$

and where $\epsilon \in \mathbb{C}$ denotes a primitive r-th root of unity. Formally,

$$W_C \in \mathbb{C}\,[x,y]_G.$$

Proof:

$$DW_C(x,y) = \sum_{i=0}^{n} A_i x^i \epsilon^i y^{n-i} = \sum_{i=0}^{n} A_i x^i y^{n-i} = W_C(x,y),$$

since $\epsilon^i = 1$ for i divisible by r, and $A_i = 0$ in the other cases. □

Finally, we show the following result for ternary codes, which is the converse of the assertion made in Exercise 1.3.20. We will use the notion of the *support* of a vector, which is just the set of coordinates where the vector is nonzero. We denote it as

$$\operatorname{supp}(v) = \{i \in n \mid v_i \neq 0\}, \qquad v \in \mathbb{F}^n.$$

In particular, $|\operatorname{supp}(u)| = \operatorname{wt}(u)$.

Lemma *Let C be a ternary linear code. Then C is 3-divisible if and only if C is self-orthogonal.* **1.6.17**

Proof: If C is self-orthogonal, then, according to Exercise 1.3.20, it is 3-divisible. Conversely, assume that C is 3-divisible. Consider two codewords u and v in C. Let $X = \operatorname{supp}(u)$ and $Y = \operatorname{supp}(v)$. Furthermore, we introduce the sets $E = \{i \mid u_i = v_i \neq 0\}$ (E for "equal") and $N = \{i \mid 0 \neq u_i \neq v_i \neq 0\}$ (N for "non-equal") and $Z = \{i \mid u_i = v_i = 0\}$ (Z for "zero"). Then E and N partition $X \cap Y$, and the sets

$$X \setminus Y, \quad Y \setminus X, \quad E, \quad N, \quad \text{and} \quad Z$$

partition the set of all coordinates. Furthermore, using the notion of the *symmetric difference* of two sets, which is defined as

$$X \Delta Y = (X \setminus Y) \,\cup\, (Y \setminus X),$$

we have for any i

$$u_i + v_i \begin{cases} \neq 0 & \text{if } i \in X\Delta Y, \\ \neq 0 & \text{if } i \in E, \\ = 0 & \text{if } i \in N, \\ = 0 & \text{if } i \in Z, \end{cases} \quad \text{and} \quad u_i - v_i \begin{cases} \neq 0 & \text{if } i \in X\Delta Y, \\ = 0 & \text{if } i \in E, \\ \neq 0 & \text{if } i \in N, \\ = 0 & \text{if } i \in Z. \end{cases}$$

Therefore, $\operatorname{wt}(u+v) = |X\Delta Y| + |E| \equiv 0 \mod 3$ and $\operatorname{wt}(u-v) = |X\Delta Y| + |N| \equiv 0 \mod 3$, so that $|E| \equiv |N| \bmod 3$. We conclude that

$$\langle u,v\rangle = \sum_i u_i v_i = \sum_{i\in E} u_i v_i + \sum_{i \in N} u_i v_i = |E| - |N| \equiv 0 \mod 3.$$

This shows that $C \subseteq C^{\perp}$. □

Exercises

E.1.6.1 **Exercise** Let C be a linear code over $\mathbb{F}_q$. Show that each $c \in C$ satisfies

$$\left|\{c' \in C \mid d(c,c') = i\}\right| = \left|\{c' \in C \mid \mathrm{wt}(c') = i\}\right| = A_i$$

for $0 \leq i \leq n$.

E.1.6.2 **Exercise** By Exercise 1.3.16, a binary self-orthogonal code is even. What about the converse?

E.1.6.3 **Exercise** Let C be a binary linear code of length n containing the all-one vector. Show that $A_i = A_{n-i}$ for $0 \leq i \leq n$.

E.1.6.4 **Exercise** Consider vectors $u, v, w \in \mathbb{F}_2^n$ satisfying $d(u,v) \equiv d(v,w) \bmod 2$. Then $d(u,w) \equiv 0 \bmod 2$. Show that this is in fact equivalent to 1.6.2.

E.1.6.5 **Exercise** Prove the following properties of one-dimensional characters of a *finite* and multiplicative group G:

- $\chi(g)$ is a $|G|$-th root of unity, i.e. $\chi(g)^{|G|} = 1_{\mathbb{F}}$.
- $\chi(g^{-1}) = \chi(g)^{-1}$. In particular, if $\mathbb{F}$ is the field $\mathbb{C}$ of complex numbers then $\chi(g^{-1}) = \overline{\chi(g)}$, where $\overline{\chi(g)}$ denotes the complex conjugate of $\chi(g)$.
- Show that the one-dimensional ordinary characters of G form a group $\hat{G}$ with respect to pointwise multiplication.

E.1.6.6 **Exercise** Let $(G,+)$ be a group. For $n \in \mathbb{Z}$ and $g \in G$ the *n-fold sum* of g is defined by

$$n \cdot g := \begin{cases} 0 & \text{if } n = 0, \\ (n-1) \cdot g + g & \text{if } n > 0, \\ (-n) \cdot (-g) & \text{if } n < 0, \end{cases}$$

where $(-g)$ is the additive inverse of g. Prove the following:

- $(n+m) \cdot g = n \cdot g + m \cdot g$ and $(nm) \cdot g = n \cdot (m \cdot g)$ for all $n, m \in \mathbb{Z}$ and $g \in G$.
- For an abelian group $(G,+)$, $n \cdot (g_1 + g_2) = n \cdot g_1 + n \cdot g_2$ for all $n \in \mathbb{Z}$, and $g_1, g_2 \in G$.
- If G is a ring then $n \cdot (g_1 g_2) = (n \cdot g_1) g_2$ is satisfied for all $n \in \mathbb{Z}$ and $g_1, g_2 \in G$.

If $(G, \cdot)$ is a multiplicative group, then, correspondingly, for $n \in \mathbb{Z}$ we use powers in order to indicate the *n-fold product* of $g \in G$ defined by

$$g^n := \begin{cases} 1 & \text{if } n = 0, \\ g^{n-1} \cdot g & \text{if } n > 0, \\ (\tilde{g})^{-n} & \text{if } n < 0, \end{cases}$$

where $\tilde{g}$ is the multiplicative inverse of g. Analogously to the n-fold sum, formulate the corresponding assertions for the n-fold product.

Exercise Recall (see e.g. [101]), that each finite abelian group G is isomorphic to a direct product of suitable cyclic groups: **E.1.6.7**

$$G \simeq \mathbb{Z}_{n_0} \times \ldots \times \mathbb{Z}_{n_{r-1}},$$

where $\prod_{i \in r} n_i = |G|$. This decomposition is unique provided that $n_i \mid n_{i+1}$ for $0 \leq i < r - 1$. In this case the number n_{r-1} is the *exponent* $\exp(G)$ of G. It is the smallest positive integer m such that the m-fold sum satisfies

$$m \cdot g = 0 \qquad \forall\, g \in G.$$

Hence, any element $g \in G$ can be written as a tuple $(g_0, \ldots, g_{r-1})$ with $g_i \in \mathbb{Z}_{n_i}$. Consider a primitive n_r-th root of unity $\xi \in \mathbb{C}$ and prove that the mapping

$$\phi \colon G \to \hat{G} \;:\; g \mapsto \left(h \mapsto \prod_{i \in r} \xi^{\frac{n_{r-1}}{n_i} g_i h_i} \right)$$

into the group of one-dimensional characters (cf. Exercise 1.6.5) is a group isomorphism, where $g = (g_0, \ldots, g_{r-1})$ and $h = (h_0, \ldots, h_{r-1})$. If we indicate the character $\phi(g)$ by χ_g, then

$$\chi_g(h) = \prod_{i \in r} \xi^{\frac{n_{r-1}}{n_i} g_i h_i} \quad \text{for each } h \in G.$$

Exercise Verify the following *orthogonality relation* for the characters of a finite abelian group G over $\mathbb{C}$: For each $g, g' \in G$ we have **E.1.6.8**

$$\frac{1}{|G|} \sum_{h \in G} \chi_{-g}(h) \chi_{g'}(h) = \begin{cases} 1 & \text{if } g = g', \\ 0 & \text{otherwise.} \end{cases}$$

Exercise Let G be a finite abelian group. We associate to each $f \colon G \to \mathbb{C}$ its *Discrete Fourier Transform* **E.1.6.9**

$$\hat{f}(h) := \sum_{g \in G} f(g) \chi_g(h), \qquad h \in G.$$

Show that

$$f(h) = \frac{1}{|G|} \sum_{g \in G} \hat{f}(g) \chi_{-g}(h), \qquad h \in G.$$

Therefore, the set $\hat{G}$ of one-dimensional characters of G forms a generating system of the vector space $\mathbb{C}^G$, whence (compare the dimension) even a basis.

E.1.6.10 **Exercise** Prove the *Lemma of Cauchy–Frobenius for Representations*: Let D denote a representation of a finite group G on a vector space over a field $\mathbb{F}$ of characteristic prime to $|G|$. Then the space

$$V_G = \{v \in V \mid \forall\, g \in G \colon\; D(g)v = v\}$$

of invariants of the group $D(G)$ is of dimension

$$\dim(V_G) = \frac{1}{|G|} \sum_{g \in G} \chi^D(g).$$

Hint: The linear mapping

$$\varphi := \frac{1}{|G|} \sum_{g \in G} D(g)$$

is a *projection,* i.e. $\varphi^2 = \varphi$.

E.1.6.11 **Exercise** Use the MacWilliams-identity in order to express $A_i^\perp := A_i(C^\perp)$ in terms of the $A_i = A_i(C)$. Rephrase your result in terms of the *Krawtchouk polynomial*

$$K_i^{n,q}(x) = \sum_{j=0}^{i} (-1)^j (q-1)^{i-j} \binom{x}{j} \binom{n-x}{i-j},$$

where

$$\binom{x}{j} := \frac{x \cdots (x-j+1)}{j!}, \; \binom{n-x}{i-j} := \frac{(n-x) \cdots (n-x-(i-j)+1)}{(i-j)!}.$$

E.1.6.12 **Exercise** Show that the parity extension of the $(7,4)$ binary Hamming-code (cf. Example 1.3.6) is self-orthogonal and hence self-dual, with minimum distance 4. Write down the MacWilliams-identity for its weight enumerator.

E.1.6.13 **Exercise** Express the coefficients of $\prod_{i \in n}(x - \kappa_i)$ in terms of elementary symmetric polynomials and the roots κ_i.

1.7 Systematic Encoding, Information Sets

1.7

It may happen that a set of k coordinates of the codewords of a fixed code always determines the remaining coordinate values. This means that if we are given the values of a codeword on those k coordinates, then the remaining $n-k$ coordinates are determined *uniquely*. We say that such a set of k coordinates forms an *information set.* The elements of an information set, i.e. the coordinates which are part of it, are called *information places.* If a k-set of coordinates is an information set, then we say that the remaining $n-k$ coordinates form a *redundancy set.* Its elements are of course called *redundancy places.* They are also called *check bits,* since they may be used for error detection and error correction.

Any code has at least one information set. It corresponds to a maximal set of columns of a generator matrix which are linearly independent. Recall that a generator matrix Γ is a $k \times n$-matrix of rank k. Such a matrix always has a set of k columns which are linearly independent. Gaussian elimination for example will reveal such a set of columns. The columns holding the pivot elements have the property that they are linearly independent. If necessary, we permute these columns up-front, for example by means of a linear isometry. This means that we may have to change to a code C' which is isometric to the original code C, which is of course no real restriction. This code C' then has the following nice property:

Corollary *Each (n,k)-code C with generator matrix Γ is linearly isometric to a code C' with generator matrix of the form* **1.7.1**

$$\Gamma' = (I_k \mid A),$$

where I_k denotes the $k \times k$-unit matrix. □

We say that a generator matrix of the form $(I_k \mid A)$ is *systematic.* The corresponding encoding map $v \mapsto v \cdot \Gamma'$ is called *systematic.* We have seen that up to linear (or semilinear) isometry, any code can be generated systematically.

When using systematic encoding $v \mapsto v \cdot \Gamma' = w$, the first k coordinate places of w simply repeat the k components of the message v. The remaining $n-k$ coordinates of w can then be used for error correction (note however, that errors may also have occurred in the first k coordinates, so decoding by simply reading out the first k coordinate values does not work). Here is an example of a generator matrix Γ and a linear isometry which determines a systematic generator matrix Γ' of a ternary code. The code generated by

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

is linearly isometric to the code generated by

$$\Gamma' := \begin{pmatrix} 0 & 2 \\ 1 & 1 \end{pmatrix} \cdot \Gamma \cdot \begin{pmatrix} 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Both are $(4,2)$-codes over $\mathbb{F}_3$.

Now we examine the effect of the linear isometries on $H(n,q) = \mathbb{F}_q^n$ that correspond to multiplication of columns by nonzero elements of the field. We want to show that such isometries map a systematic code onto a systematic one.

Isometries obtained by multiplications are described by regular $n \times n$ diagonal matrices D, and these matrices form a normal subgroup $D_n(q)$ of $M_n(q)$ (see Exercise 1.7.4). An (n,k)-code C with generator matrix $(I_k \mid A)$, where A is the $k \times (n-k)$-matrix

$$A = \begin{pmatrix} a_{0,k} & \dots & a_{0,n-1} \\ \vdots & & \vdots \\ a_{k-1,k} & \dots & a_{k-1,n-1} \end{pmatrix},$$

is mapped under that kind of isometries onto an isometric code C' with generator matrix

$$(I_k \mid A) \cdot D.$$

Any matrix which can be obtained from the above generator matrix via left multiplication by a regular $k \times k$-matrix is a generator matrix of C' as well. Suppose we choose for the left multiplication the upper left part of the multiplicative inverse of $D = (d_{ij}) \in D_n(q)$, i.e. the matrix

$$D' := \begin{pmatrix} d_{0,0}^{-1} & & 0 \\ & \ddots & \\ 0 & & d_{k-1,k-1}^{-1} \end{pmatrix},$$

then we obtain the systematic generator matrix

$$D' \cdot (I_k \mid A) \cdot D = (I_k \mid D * A)$$

of C', where

1.7.2 $$D * A := \left(d_{ii}^{-1} a_{ij} d_{jj} \right)_{0 \le i < k, k \le j < n}.$$

This proves the following

Lemma *For each $D \in D_n(q)$, the systematic matrices $(I_k \mid A)$ and $(I_k \mid D * A)$ generate linearly isometric codes.* □ **1.7.3**

Another characterization of information sets is the following. Let J be a set of column indices, i.e. $J \subseteq \{0, \ldots, n-1\} = n$. Denote the complement of J as

$$\overline{J} := n \setminus J.$$

Then

$$\mathbb{F}_q^{(J)} := \left\{ (w_0, \ldots, w_{n-1}) \in \mathbb{F}_q^n \;\middle|\; w_j = 0 \text{ for all } j \in \overline{J} \right\} \tag{1.7.4}$$

is a subspace of $\mathbb{F}_q^n$ of dimension $|J|$. In particular,

$$\mathbb{F}_q^{(J)} \oplus \mathbb{F}_q^{(\overline{J})} = \mathbb{F}_q^n. \tag{1.7.5}$$

Theorem *An (n,k)-code C possesses a k-subset $J \subseteq n$ as an information set if and only if $C \oplus \mathbb{F}_q^{(\overline{J})} = \mathbb{F}_q^n$ holds true.* □ **1.7.6**

The proof of this theorem is Exercise 1.7.7. Since each element of $\mathbb{F}_q^{(J)}$ is of Hamming weight at most $|J|$, we obtain the following

Theorem *Consider $d \in \mathbb{N}^*$. For each linear code C of length n over $\mathbb{F}_q$, the following conditions are equivalent:* **1.7.7**

- *C has minimum weight at least d.*
- *For each $J \subseteq n$, where $|J| < d$, we have*

$$C \cap \mathbb{F}_q^{(J)} = \{0\}\,. \qquad \square$$

Now we want to describe the close connection between systematic generator matrices and systematically encoded linear codes.

Theorem *Assume that $1 \le k \le n-1$. The mapping* **1.7.8**

$$A \mapsto \left\{ v \cdot (I_k \mid A) \;\middle|\; v \in \mathbb{F}_q^k \right\}$$

is a bijection between the set of $k \times (n-k)$-matrices A over $\mathbb{F}_q$ and the set of systematically encoded (n,k)-codes over $\mathbb{F}_q$.

Proof: The given mapping is obviously surjective. In order to prove injectivity, we consider two $k \times (n-k)$-matrices A and B over $\mathbb{F}_q$ which differ in their i-th row for some i. If $e^{(i)}$ denotes the i-th unit vector, then the codewords $e^{(i)} \cdot (I_k \mid A)$ and $e^{(i)} \cdot (I_k \mid B)$ are distinct. However, the two codewords agree in all of the first k coordinates, whose values by Theorem 1.7.6 determine the codeword uniquely. The only possibility for this is that the codes generated by $(I_k \mid A)$ and $(I_k \mid B)$ are distinct. This proves the statement. □

When classifying linear codes we want to obtain complete lists of representatives of the isometry classes for given parameters n, k and q. It is most convenient to describe the representatives by systematic generator matrices. Therefore, it is of interest to determine all systematic generator matrices of codes belonging to a single isometry class.

1.7.9 **Remark** [184, 2.10] Assume that $\Gamma = (I_k \mid A)$ is a systematic generator matrix of an (n,k)-code over $\mathbb{F}_q$ with $k < n$. The systematic generator matrices of codes (semi)linearly isometric to C can be obtained as follows. Apply a (semi)linear isometry so that the first k columns of the resulting matrix Γ' are linearly independent. Then pre-multiply Γ' by a suitable matrix from $\mathrm{GL}_k(q)$. There are several types of isometry operations which guarantee that the first k columns of Γ' are linearly independent. They can be generated by repeated application of the following isometry operations:

- Considering permutational isometries we obtain: The permutations of columns that replace the first k columns of Γ by linearly independent columns can be generated by repeated application of three types of permutations:
  1. Interchange the columns with index i and j, where $i,j < k$. After interchanging the i-th and j-th row of Γ', the resulting matrix is again systematic.
  2. Interchange the columns with index i and j, where $i,j \geq k$, then the resulting matrix is systematic.
  3. Interchange the columns with index i and j, where $i < k \leq j$. This is only possible in case $a_{ij} \neq 0$, for otherwise the first k columns of Γ' would no longer be linearly independent. In order to obtain a systematic matrix, multiply the i-th row of Γ' by a_{ij}^{-1}, and for $\ell \neq i$ subtract this new row multiplied by $a_{\ell j}$ from the ℓ-th row of Γ'.
- Furthermore, using linear isometries it is possible to multiply columns of Γ by nonzero field elements. If we multiply the i-th column of Γ by $\kappa \in \mathbb{F}_q^*$, say, then the resulting matrix either is already systematic (namely if $i \geq k$), or can be brought into a systematic form (namely by multiplying the i-th row by κ^{-1}).
- When considering also semilinear isometries, apply an automorphism $\alpha \in \mathrm{Aut}(\mathbb{F}_q)$ to each entry of Γ. The resulting matrix is again systematic. ◇

Exercises

E.1.7.1 **Exercise** Use the existence of systematic generator matrices in order to show that each linear code with $k = n - 1$ has a minimum distance at most 2.

Exercise Let C be an (n, k)-code with minimum distance d. Show that d is the largest integer with the property that any $n - d + 1$ coordinate positions contain an information set. **E.1.7.2**

Exercise There are three types of elementary $k \times k$-matrices over $\mathbb{F}$. For $\lambda \in \mathbb{F}^*$ and for $i_0, j_0 \in k$ with $i_0 \neq j_0$ they are given by: **E.1.7.3**

- $B^{(1)}_{i_0,\lambda}$ is the unit matrix I_k in which the entry 1 occurring in position (i_0, i_0) is replaced by λ, thus it is a diagonal matrix $(b_{ij})_{i,j \in k}$ with

$$b_{ij} = \begin{cases} \lambda & \text{if } i = j = i_0, \\ 1 & \text{if } i = j \neq i_0, \\ 0 & \text{else.} \end{cases}$$

- $B^{(2)}_{i_0,j_0,\lambda}$ is the unit matrix I_k with an additional entry λ in position (i_0, j_0), thus it is the matrix $(b_{ij})_{i,j \in k}$ with

$$b_{ij} = \begin{cases} 1 & \text{if } i = j, \\ \lambda & \text{if } i = i_0 \text{ and } j = j_0, \\ 0 & \text{else.} \end{cases}$$

- $B^{(3)}_{i_0,j_0}$ is the unit matrix I_k in which the rows (or columns) of index i_0 and j_0 are exchanged, thus it is the matrix $(b_{ij})_{i,j \in k}$ with

$$b_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } (i \neq i_0 \text{ or } j \neq j_0), \\ 1 & \text{if } (i,j) = (i_0, j_0) \text{ or } (i,j) = (j_0, i_0), \\ 0 & \text{else.} \end{cases}$$

Prove that all these matrices are regular, and that the inverse of an elementary matrix is again elementary. Deduce then that every matrix of $\mathrm{GL}_k(q)$ can be written as a product of elementary matrices.

Show that the following holds true: Multiplying a $k \times n$-matrix Γ from the left with an elementary matrix B yields an elementary row operation on Γ. Hence, $B \cdot \Gamma$ is a composition of elementary row operations on Γ for all $B \in \mathrm{GL}_k(q)$. Multiplying an $n \times k$-matrix Γ from the right with an elementary matrix yields an elementary column operation on Γ. Hence, $\Gamma \cdot B$ is a composition of elementary column operations on Γ for all $B \in \mathrm{GL}_k(q)$.

Exercise Check that the regular $n \times n$ diagonal matrices over $\mathbb{F}_q$ form a *normal subgroup* $D_n(q)$ of $M_n(q)$, which means that $D_n(q)$ is a subgroup of $M_n(q)$ and that $M^{-1} \cdot D \cdot M \in D_n(q)$ for each $D \in D_n(q)$ and $M \in M_n(q)$. **E.1.7.4**

E.1.7.5 **Exercise** Verify that the composition $*$ of 1.7.2 satisfies

$$D_1 * (D_2 * A) = (D_1 \cdot D_2) * A \text{ and } I_n * A = A.$$

Here, D_1 and D_2 are elements of $D_n(q)$ and A is any $k \times k$-matrix. In particular, this operation is a group action.

E.1.7.6 **Exercise** Prove that 1.7.5 holds for all subsets $J \subseteq n$.

E.1.7.7 **Exercise** Prove 1.7.6.

E.1.7.8 **Exercise** Assume that W and W' are subspaces of the vector space V. Prove that the following two statements are equivalent:

1. $V = W \oplus W'$ (which means $V = W + W'$ and $W \cap W' = \{0\}$).
2. For each $v \in V$ there exist uniquely determined $w \in W$ and $w' \in W'$ such that $v = w + w'$.

E.1.7.9 **Exercise** Assume that W and W' are subspaces of the finite dimensional vector space V. Prove that the following two statements are equivalent:

1. $V = W \oplus W'$.
2. $V = W + W'$ and $\dim(V) = \dim(W) + \dim(W')$.

1.8

1.8 A Minimum Distance Algorithm

As we have seen, the minimum distance is a very important parameter of a linear code. Nevertheless, *evaluating* this parameter for a given code may turn out to be surprisingly hard. As example 1.3.8 shows, the minimum distance of a code can be less than the minimum weight of the rows of a particular generator matrix. Here we present an algorithm, which is a variation of an idea of A. Brouwer and due to K.-H. Zimmermann. Information sets play an important role in this algorithm. It uses an iteration of Gaussian elimination and it works efficiently if the code under consideration has many information sets which are pairwise disjoint.

Algorithm (MinDist) To compute the minimum distance of a given linear (n,k)-code C. The *input* is a generator matrix Γ of C and the *output* is the minimum distance $d = \mathrm{dist}(C)$ of C. **1.8.1**

- Recall that the code in question is linearly isometric to a systematic one. This means that there exist matrices $M \in M_n(q)$ and $B \in \mathrm{GL}_k(q)$ such that $B \cdot \Gamma \cdot M^\top$ is a systematic generator matrix. In fact, recalling the Gaussian Algorithm, we can obtain a systematic generator matrix by elementary row operations, i.e. by multiplying from the left with a matrix $B_1 \in \mathrm{GL}_k(q)$, and a suitable column permutation, i.e. a multiplication from the right by the transpose of a permutation matrix $M_{\pi_1} := M_{(\epsilon;\pi_1)}$ (cf. 1.4.8),
$$\Gamma_1 := B_1 \cdot \Gamma \cdot M_{\pi_1}^\top = (\, I_{k_1} \mid A_1),$$
where $k_1 = k$.

- If A_1 is neither empty nor a zero matrix, its rank is k_2 with $0 < k_2 \leq k_1$. Applying Gaussian elimination, we can obtain k_2 different unit vectors in the remaining $n - k_1$ columns. Of course, this process may distort the original unit matrix I_{k_1} in the leftmost k_1 columns. In other words, we can multiply Γ_1 from the left by an element B_2 of $\mathrm{GL}_k(q)$ and from the right by the transpose of a permutation matrix M_{π_2} with $\pi_2(j) = j$ for $0 \leq j < k_1$, obtaining a generator matrix of a linearly isometric code,
$$\Gamma_2 := B_2 \cdot \Gamma_1 \cdot M_{\pi_2}^\top = \left(\begin{array}{c|c|c} A_2' & I_{k_2} & A_2 \\ \cline{2-3} & 0 & 0 \end{array} \right).$$
The matrix A_2' is a $k \times k_1$-matrix and A_2 is a $k_2 \times (n - k_1 - k_2)$-matrix. The zeros indicate zero matrices.

- Assume that for $i \geq 2$ the matrix A_i which has just been computed is neither empty nor a zero matrix. Then its rank is k_{i+1} with $0 < k_{i+1} \leq k_i$. We continue this way, obtaining regular matrices $B_{i+1} \in \mathrm{GL}_k(q)$, permutation matrices $M_{\pi_{i+1}} \in M_n(q)$, with $\pi_{i+1}(j) = j$ for $0 \leq j < k_1 + \ldots + k_i$, and generator matrices
$$\Gamma_{i+1} := B_{i+1} \cdot \Gamma_i \cdot M_{\pi_{i+1}}^\top = \left(\begin{array}{c|c|c} A_{i+1}' & I_{k_{i+1}} & A_{i+1} \\ \cline{2-3} & 0 & 0 \end{array} \right),$$
A_{i+1}' a $k \times (k_1 + \ldots + k_i)$-matrix and A_{i+1} a $k_{i+1} \times (n - k_1 - \ldots - k_{i+1})$-matrix. We repeat this procedure. Eventually, we will obtain a generator matrix Γ_m, say, such that
$$\Gamma_m := B_m \cdot \Gamma_{m-1} \cdot M_{\pi_m}^\top = \left(\begin{array}{c|c|c} A_m' & I_{k_m} & A_m \\ \cline{2-3} & 0 & 0 \end{array} \right),$$

where A_m is either empty (which means it has no columns) or it is a zero matrix. Then, $k_1 + \ldots + k_m \leq n$ and A_m has $n - k_1 - \ldots - k_m$ columns. Consequently, the generator matrix Γ_m has $n - k_1 - \ldots - k_m$ zero columns, whence all elements of the code generated by Γ_m have weight at most $k_1 + \ldots + k_m$.

- Let $\widetilde{C}$ be the code generated by Γ_m. We note that C is linearly isometric to this code, whereas the matrices

$$\Gamma_1, \ldots, \Gamma_m$$

generate codes which are *linearly isometric* to $\widetilde{C}$ but not necessarily *equal* to $\widetilde{C}$ (except for Γ_m, of course). For this reason we put

$$\widetilde{\Gamma}_i := \Gamma_i \cdot M_{\pi_{i+1}}^{\top} \cdots M_{\pi_m}^{\top} = B_i \cdots B_1 \cdot \Gamma \cdot M_{\pi_1}^{\top} \cdots M_{\pi_m}^{\top} = \widetilde{B}_i \Gamma_m,$$

$\widetilde{B}_i \in \mathrm{GL}_k(q)$, so that the matrices

$$\widetilde{\Gamma}_1, \ldots, \widetilde{\Gamma}_m$$

generate the *same* code $\widetilde{C}$. Moreover, the leftmost $k_1 + \ldots + k_i$ columns of Γ_i and $\widetilde{\Gamma}_i$ are the same, whence $\widetilde{\Gamma}_i$ has the unit matrix I_{k_i} in the same position as Γ_i.

- Using these matrices, we define for $1 \leq i \leq k$ the following subsets $\widetilde{C}_i$ of C:

$$\widetilde{C}_i := \bigcup_{j=1}^{m} \left\{ v \cdot \widetilde{\Gamma}_j \mid v \in \mathbb{F}_q^k,\ \mathrm{wt}(v) \leq i \right\}.$$

Clearly, these sets form the ascending chain

$$\widetilde{C}_1 \subseteq \widetilde{C}_2 \subseteq \ldots \subseteq \widetilde{C}_k = \widetilde{C}$$

of subsets of $\widetilde{C}$, and hence the minimum weights

$$\overline{d}_i := \min \left\{ \mathrm{wt}(c) \mid c \in \widetilde{C}_i,\ c \neq 0 \right\},$$

form the decreasing sequence

$$\overline{d}_1 \geq \overline{d}_2 \geq \ldots \geq \overline{d}_k = \mathrm{dist}(\widetilde{C}) = \mathrm{dist}(C).$$

In most cases, we do not need to compute all of these values. In fact, the computation of $\overline{d}_k$ is just the evaluation of $\mathrm{dist}(\widetilde{C})$ as the least weight of all codewords c in $\widetilde{C} \setminus \{0\}$, which we want to avoid, if possible. As a matter of fact, in the first step we just compute $\overline{d}_1$. Later, if $\overline{d}_i$ has been computed for some $i \geq 1$, we will compare it with $\underline{d}_i$, which is a lower bound for the weight of the elements in $\widetilde{C} \setminus \widetilde{C}_i$. If $\overline{d}_i \leq \underline{d}_i$ we are finished. Otherwise, if $\overline{d}_i > \underline{d}_i$, we proceed to compute the exact value of $\overline{d}_{i+1}$.

- Hence, we try to find lower bounds for the weights in the complements $\widetilde{C} \setminus \widetilde{C}_i$. For this purpose we pick an element $c \in \widetilde{C} \setminus \widetilde{C}_i$. Since $c \notin \widetilde{C}_i$, there exists, for each j, a vector $v^{(j)} \in \mathbb{F}_q^k$ such that
$$c = v^{(j)} \cdot \widetilde{\Gamma}_j, \quad 1 \leq j \leq m, \text{ and } \operatorname{wt}(v^{(j)}) \geq i+1.$$
In order to estimate the weight of c, we consider each of these representations of c by using the various information places in $\widetilde{\Gamma}_j$, the columns of which contain the unit matrix I_{k_j}. These are the columns of index r for $k_1 + \ldots + k_{j-1} \leq r < k_1 + \ldots + k_j$. We are especially interested in the k_j coordinates c_r of $c = v^{(j)} \cdot \widetilde{\Gamma}_j$ corresponding to these k_j columns. Since $v^{(j)}$ is of length k, these entries of c contribute at least the value $i + 1 - (k - k_j)$ to the weight of c. Since these sets of places are disjoint, for different j, we obtain
$$\operatorname{wt}(c) \geq \sum_{j=1}^{m} (i + 1 - (k - k_j)).$$
We can restrict our attention to positive summands, which gives the lower bound
$$\operatorname{wt}(c) \geq \sum_{j:k-k_j \leq i} (i + 1 - (k - k_j)) =: \underline{d}_i.$$
Obviously, since the first summand is $i + 1$, the sequence of these bounds is increasing:
$$2 \leq \underline{d}_1 < \underline{d}_2 < \underline{d}_3 < \ldots < \underline{d}_k.$$
In addition,
$$\underline{d}_k = m + \sum_{j=1}^{m} k_j.$$

- Since $\operatorname{wt}(c) \leq k_1 + \ldots + k_m$ for all $c \in \widetilde{C}$, there exists a smallest index i_0 such that
$$\overline{d}_{i_0} \leq \underline{d}_{i_0}.$$
For this i_0 we have that
$$\overline{d}_{i_0} := \min \left\{ \operatorname{wt}(c) \mid c \in \widetilde{C}_{i_0},\ c \neq 0 \right\},$$
and the inequality
$$\underline{d}_{i_0} \leq \min \left\{ \operatorname{wt}(c) \mid c \in \widetilde{C} \setminus \widetilde{C}_{i_0} \right\}$$
holds true. Hence,
$$\overline{d}_{i_0} = \operatorname{dist}(\widetilde{C}),$$
and a codeword of weight $\operatorname{dist}(\widetilde{C})$ is contained in $\widetilde{C}_{i_0}$.

- To simplify the algorithm, it is possible to do all these computations with Γ_i instead of $\widetilde{\Gamma}_i$. Notice that the values of $\overline{d}_j$ and d_j do not change when we use Γ_i instead of $\widetilde{\Gamma}_i$ since

$$\operatorname{wt}\left(v \cdot \widetilde{\Gamma}_i\right) = \operatorname{wt}\left(v \cdot \Gamma_i \cdot M_{\pi_{i+1}}^{\top} \cdots M_{\pi_m}^{\top}\right) = \operatorname{wt}(v \cdot \Gamma_i).$$

Moreover, we can replace the sets $\widetilde{C}_i$ by the *isometric sets*

$$C_i := \bigcup_{j=1}^{m} \left\{ v \cdot \Gamma_j \mid v \in \mathbb{F}_q^k,\ \operatorname{wt}(v) \leq i \right\}. \qquad \square$$

Here is a summary of the algorithm **MinDist**.

1.8.2 **Algorithm** Compute the minimum distance of a given linear (n, k)-code C.

Input: A systematic generator matrix $\Gamma_1 = (I_k \mid A_1)$ of C.

Output: The minimum distance $\operatorname{dist}(C)$.

```
$m := 2$
$k_1 := k$
repeat
    Apply Gaussian elimination and possibly permutations of the
    columns to the matrix $A_{m-1}$ from $\Gamma_{m-1} = \left(\begin{array}{c|c|c} A'_{m-1} & I_{k_{m-1}} & A_{m-1} \\ \cline{2-3} & 0 & 0 \end{array}\right)$
    to obtain a generator matrix $\Gamma_m = \left(\begin{array}{c|c|c} A'_m & I_{k_m} & A_m \\ \cline{2-3} & 0 & 0 \end{array}\right)$
until $\operatorname{rank}(A_m) = 0$
$C_0 := \{0\}$
i := 0
repeat
    $i := i + 1$
    $C_i := C_{i-1} \cup \bigcup_{j=1}^{m} \{v \cdot \Gamma_j \mid v \in \mathbb{F}(q)^k, \operatorname{wt}(v) = i\}$
    $\overline{d_i} := \min\{\operatorname{wt}(c) \mid c \in C_i, c \neq 0\}$
    $\underline{d_i} := \sum_{\substack{j=1 \\ k-k_j \leq i}}^{m} (i+1) - (k - k_j)$
until $\overline{d_i} \leq \underline{d_i}$
return $\overline{d_i}$
```

□

Example We apply the algorithm **MinDist** to the binary $(7,3)$-code C with generator matrix 1.8.3

$$\Gamma_1 = \left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right).$$

This matrix has the information set $\{0,1,2\}$. The algorithm successively computes the generator matrices

$$\Gamma_2 = \left(\begin{array}{ccc|ccc|c} 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right)$$

with information set $\{3,4,5\}$ and the generator matrix

$$\Gamma_3 = \left(\begin{array}{cccccc|c} 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right)$$

with information set $\{6\}$. The set C_1 consists of all rows of the three generator matrices Γ_1, Γ_2 and Γ_3. Each of them is of weight 4, whence $\overline{d}_1 = 4$. The lower bound for the minimum weight of the vectors outside of C_1 is $\underline{d}_1 = 4$. Hence, $d = \overline{d}_1 = 4$ is the minimum distance of C. ◇

Example We apply the algorithm **MinDist** to the binary $(15,5)$-code C with generator matrix 1.8.4

$$\Gamma = \left(\begin{array}{ccccccccccccccc} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}\right).$$

This code will be constructed in 4.3.5, it is a BCH-code. The systematic matrices are

$$\Gamma_1 = \left(\begin{array}{ccccc|cccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}\right),$$

$$\Gamma_2 = \left(\begin{array}{ccccc|ccccc|ccccc} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right),$$

and

$$\Gamma_3 = \left(\begin{array}{cccccccccc|ccccc} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right).$$

The information sets are

$$\{0,1,2,3,4\},\ \{5,6,7,8,9\},\ \text{and}\ \{10,11,12,13,14\}.$$

The minimum weight of the rows in these matrices is $\overline{d}_1 = 7$, whereas $\underline{d}_1 = 2-(5-5)+2-(5-5)+2-(5-5) = 6$. Since $7 > 6$, we continue by considering linear combinations of any two rows of the Γ_i. For example, if $i = 1$ we look at vectors v and codewords $v \cdot \Gamma_1$ where

v	$v \cdot \Gamma_1$	$\mathrm{wt}(v \cdot \Gamma_1)$
(1,1,0,0,0)	(1,1,0,0,0,0,1,0,1,0,0,1,1,0,1)	7
(1,0,1,0,0)	(1,0,1,0,0,1,1,0,1,1,1,0,0,0,0)	7
(1,0,0,1,0)	(1,0,0,1,0,0,0,1,1,1,1,0,1,0,1)	8
(1,0,0,0,1)	(1,0,0,0,1,1,1,1,0,1,0,1,1,0,0)	8
(0,1,1,0,0)	(0,1,1,0,0,1,0,0,0,1,1,1,1,0,1)	8
(0,1,0,1,0)	(0,1,0,1,0,0,1,1,0,1,1,1,0,0,0)	7
(0,1,0,0,1)	(0,1,0,0,1,1,0,1,1,1,0,0,0,0,1)	7
(0,0,1,1,0)	(0,0,1,1,0,1,1,1,0,0,0,0,1,0,1)	7
(0,0,1,0,1)	(0,0,1,0,1,0,0,1,1,0,1,1,1,0,0)	7
(0,0,0,1,1)	(0,0,0,1,1,1,1,0,1,0,1,1,0,0,1)	8

If $i = 2$, we have

v	$v \cdot \Gamma_2$	$\mathrm{wt}(v \cdot \Gamma_2)$
(1,1,0,0,0)	(0,1,1,0,1,1,1,0,0,0,0,1,0,1,0)	7
(1,0,1,0,0)	(1,0,0,0,0,1,0,1,0,0,1,1,0,1,1)	7
(1,0,0,1,0)	(1,0,1,0,1,1,0,0,1,0,0,0,1,1,1)	8
(1,0,0,0,1)	(0,1,1,0,0,1,0,0,0,1,1,1,1,0,1)	8
(0,1,1,0,0)	(1,1,1,0,1,0,1,1,0,0,1,0,0,0,1)	8
(0,1,0,1,0)	(1,1,0,0,0,0,1,0,1,0,0,1,1,0,1)	7
(0,1,0,0,1)	(0,0,0,0,1,0,1,0,0,1,1,0,1,1,1)	7
(0,0,1,1,0)	(0,0,1,0,1,0,0,1,1,0,1,1,1,0,0)	7
(0,0,1,0,1)	(1,1,1,0,0,0,0,1,0,1,0,0,1,1,0)	7
(0,0,0,1,1)	(1,1,0,0,1,0,0,0,1,1,1,1,0,1,0)	8

and for $i = 3$ we obtain

v	$v \cdot \Gamma_3$	$\mathrm{wt}(v \cdot \Gamma_3)$
$(1,1,0,0,0)$	$(0,1,0,1,0,0,1,1,0,1,1,1,0,0,0)$	7
$(1,0,1,0,0)$	$(1,1,0,1,1,1,0,0,0,0,1,0,1,0,0)$	7
$(1,0,0,1,0)$	$(0,0,1,1,1,1,0,1,0,1,1,0,0,1,0)$	8
$(1,0,0,0,1)$	$(1,1,1,0,1,0,1,1,0,0,1,0,0,0,1)$	8
$(0,1,1,0,0)$	$(1,0,0,0,1,1,1,1,0,1,0,1,1,0,0)$	8
$(0,1,0,1,0)$	$(0,1,1,0,1,1,1,0,0,0,0,1,0,1,0)$	7
$(0,1,0,0,1)$	$(1,0,1,1,1,0,0,0,0,1,0,1,0,0,1)$	7
$(0,0,1,1,0)$	$(1,1,1,0,0,0,0,1,0,1,0,0,1,1,0)$	7
$(0,0,1,0,1)$	$(0,0,1,1,0,1,1,1,0,0,0,0,1,0,1)$	7
$(0,0,0,1,1)$	$(1,1,0,1,0,1,1,0,0,1,0,0,0,1,1)$	8

This shows that $\overline{d}_2 = 7$. On the other hand, $\underline{d}_2 = 3-(5-5)+3-(5-5)+3-(5-5) = 9$ which is greater than 7, i.e. the minimum distance has been determined to be 7. In this example, we have looked at $15 + 3 \cdot 10 = 45$ codewords, which is actually worse than the original problem. ◇

We see that the algorithm may actually be worse than the original problem. But in many cases, in particular when the codes get bigger, there is a benefit. For example, the minimum distance of the binary extended Golay code of length 24 and dimension 12 (presented in 2.3.12) is computed by looking at 596 rather than $2^{12} = 4096$ codewords.

Exercises

Exercise Prove the remaining statements about $\underline{d}_i$ for $1 \leq i \leq k$ in the description of 1.8.1. **E.1.8.1**

Exercise Use the algorithm **MinDist** in order to evaluate the minimum distance of the binary $(7,4)$-code with generator matrix **E.1.8.2**

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

Check your result using the attached software.