

Simeon Ball

A Course in Algebraic Error-Correcting Codes

Compact Textbooks in Mathematics

Compact Textbooks in Mathematics

This textbook series presents concise introductions to current topics in mathematics and mainly addresses advanced undergraduates and master students. The concept is to offer small books covering subject matter equivalent to 2- or 3-hour lectures or seminars which are also suitable for self-study. The books provide students and teachers with new perspectives and novel approaches. They may feature examples and exercises to illustrate key concepts and applications of the theoretical contents. The series also includes textbooks specifically speaking to the needs of students from other disciplines such as physics, computer science, engineering, life sciences, finance.

- **compact:** small books presenting the relevant knowledge
- **learning made easy:** examples and exercises illustrate the application of the contents
- **useful for lecturers:** each title can serve as basis and guideline for a semester course/lecture/seminar of 2–3 hours per week.

More information about this series at <http://www.springer.com/series/11225>

Simeon Ball

A Course in Algebraic Error-Correcting Codes

 Birkhäuser

Simeon Ball
Department of Mathematics
Polytechnic University of Catalonia
Barcelona, Spain

ISSN 2296-4568 ISSN 2296-455X (electronic)
Compact Textbooks in Mathematics
ISBN 978-3-030-41152-7 ISBN 978-3-030-41153-4 (eBook)
<https://doi.org/10.1007/978-3-030-41153-4>

Mathematics Subject Classification (2010): 94BXX, 51EXX, 94AXX

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This book is published under the imprint Birkhäuser, www.birkhauser-science.com by the registered company Springer Nature Switzerland AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book is based on lecture notes for a course on coding theory given as part of the Applied Mathematics and Mathematical Engineering master's degree at the Universitat Politècnica de Catalunya. The aim of the course is to give an up-to-date account of error-correcting codes from a mathematical point of view, an analysis of the construction of these codes, and of the various algorithms which are implemented to correct corrupted messages.

The lectures were prepared for an audience at master's level, although a large proportion of the book should be accessible to students at undergraduate level and to engineering and physics students too. There is some formal algebra that may not be familiar, mainly in the introduction of finite fields in Chapter 2, but this is not essential to be able to follow the main part of the content. It is enough to know how to perform finite field arithmetic and how to factorise polynomials over finite fields, all of which are explained in detail in Chapter 2.

A large part of the material included in the text dates back to the latter part of the last century. However, there have been recent advances in the algebraic theory of error-correcting codes, many of which are included here. There are still questions and problems which remain unresolved despite considerable effort having been directed towards their resolution. Many of these are highlighted in the text. The book takes a combinatorial, algebraic, and geometric view of coding theory but combines this with a practical consideration of how the codes constructed are implemented.

Shannon's theorem, the highlight of Chapter 1 and which dates back to 1947, tells us that given a noisy, not totally unreliable, communication channel, there are codes which provide a means of reliable communication at a transmission rate arbitrarily close to the capacity. However, Shannon's theorem only tells us that reliable communication is possible, it does not provide us with a feasible way in which to encode or decode the transmission. One of the aims of the book is to find codes with a high transmission rate, which allow fast encoding and decoding. On the way towards this objective, we construct various types of codes and consider a number of decoding algorithms.

Chapter 2 is a brief introduction to finite fields and the geometries associated with these fields. An emphasis is given to the factorisation of cyclotomic polynomials over finite fields, which is put to use in Chapter 5.

The concept of a minimum distance between any two codewords of a block code is introduced in Chapter 3. The larger the minimum distance, the more errors one can correct and, together with the length and size of the code, are the fundamental parameters of a block code. There are various bounds on these parameters proven in Chapter 3, both the Gilbert–Varshamov lower bound, given by the greedy algorithm, and upper bounds, a discrete

sphere-packing bound and the better Plotkin and Elias-Bassalygo bounds. The codes given by the Gilbert–Varshamov bound are asymptotically good codes in the sense that both the transmission rate, the proportion of the bits of the message which contains information, and the relative minimum distance are bounded away from zero as the length of the code increases. However, in practice, as in the case of the randomly chosen code, there is no efficient way in which to encode or decode the message using these codes.

The advantage of linear codes, the focus of Chapter 4 and fundamental to most of the rest of the book, is that they are efficient to encode. One can encode by simply multiplying a vector by a matrix. We consider a decoding algorithm for linear codes based on syndromes. The question of existence of a vector of small weight with a specified syndrome is shown to be NP-complete, which implies that the decoding algorithm is not feasible for long codes, although it is used in practice for short length codes.

The classical 2-error correcting and 3-error correcting perfect codes are constructed in Chapter 5, as well as the general class of BCH codes. Although BCH codes exist for all lengths, it is known that there are no sequences of BCH codes for which the transmission rate and the relative minimum distance are both bounded away from zero.

Reed–Solomon codes are codes whose codewords are the evaluation of polynomials of low degree. In Chapter 6, we will exploit this algebraic structure to explicitly develop a polynomial time decoding algorithm for Reed–Solomon codes which will correct any error bits, providing the number of errors is less than half the minimum distance. We will also show that there is a polynomial time list decoding algorithm which produces a short list of possible candidates for the sent codeword when far more errors occur. By employing two Reed–Solomon codes, this allows one to decode correctly well beyond the half the minimum distance bound with very high probability. Another important application of MDS codes, and in particular Reed–Solomon codes, is the storage of data in distributed storage systems. The fact that the codewords of an MDS code are uniquely determined by relatively few bits means that data, stored across a number of different servers, can be recovered from just a few.

In Chapter 7 we prove that there are subfield subcodes of generalised Reed–Solomon codes meeting the Gilbert–Varshamov bound. Since these codes are linear, they are fast to encode and the fact that they have an algebraic structure allows us to decode using the list decoding algorithm from Chapter 6. We then go on to consider codes constructed from algebraic curves. These algebraic-geometric codes include codes which surpass the Gilbert–Varshamov bound for codes over large alphabets.

There are linear codes constructed from low-density parity check matrices for which both the transmission rate and the relative minimum distance are bounded away from zero. We will prove in Chapter 8 that we can encode and decode certain low-density parity check codes with polynomial time

algorithms. These codes are widely implemented in wireless communication, with a transmission rate close to the capacity set out by Shannon's theorem.

Although not asymptotically good, Reed–Muller codes and their subcodes, the theme of Chapter 9, are widely implemented since there are fast decoding algorithms for these codes, such as a majority logic decoding algorithm which is detailed here. Kerdock codes are certain subcodes of the second-order Reed–Muller codes. They are of particular interest since there are Kerdock codes which are non-linear codes with parameters for which it is known that no linear code exists.

Bringing together p -adic numbers and cyclic codes in Chapter 10, we construct non-linear codes which are linear over the ring of integers modulo a prime power. Within this class of codes we again construct a non-linear binary code with parameters for which it is known no binary linear code exists. This suggests that, more generally, these codes could be a source of codes which perform better than linear codes.

The three main conjectures concerning error-correcting codes are included. The *Information Theory and Applications Center* of the *University of California San Diego* offers prizes for the resolution of any of these three conjectures. The three conjectures can be roughly stated as, there is no infinite sequence of binary codes better than the Gilbert–Varshamov bound, there are no non-trivial constant weight perfect codes, and there are no linear MDS codes longer than the Reed–Solomon codes, apart from some three-dimensional even characteristic codes and their duals.

I would like to thank Tim Alderson, Anurag Bishnoi, Aart Blokhuis, Massimo Giulietti, Victor Hernandez, Michel Lavrauw, Sonia Mansilla, Valentina Pepe, and Oriol Serra for their comments and suggestions. I would also like to thank Francesc Comellas for drawing some of the figures.

Barcelona, Spain
October 2019

Simeon Ball

The dependencies between the chapters are as follows.

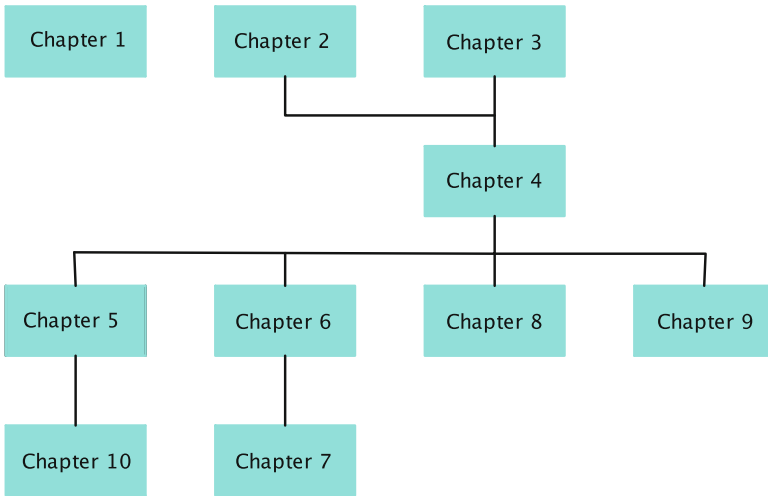


Table of Parameters for Codes in the Text

The following table lists the parameters of the specific codes constructed in the text. An $[n, k, d]_q$ code refers to a k -dimensional linear code over \mathbb{F}_q of length n and minimum distance d . A $(n, K, d)_r$ code refers to a code of length n , size K , and minimum distance d over an alphabet of size r .

	Parameters	Name
Example 3.4	$(6, 8, 3)_2$	
Example 3.6	$(7, 8, 4)_2$	
Example 3.11	$(6, 4, 4)_2$	
Exercise 3.2	$(7, 16, 3)_2$	
Exercise 3.4	$(10, 6, 6)_2$	
Exercise 3.5	$(6, 4, 5)_3$	
Example 4.2	$[7, 4, 3]_2$	Binary Hamming
Example 4.3	$[9, 3, 6]_3$	
Example 4.5	$[(q^m - 1)/(q - 1), (q^m - 1)/(q - 1) - m, 3]_q$	Hamming
Example 4.8	$[8, 4, 4]_3$	
Example 4.16	$[q\sqrt{q} + 1, 3, q\sqrt{q} - \sqrt{q}]_q$	Hermitian curve
Example 4.19	$[10, 6, 4]_3$	
Example 4.23	$[10, 4, 6]_3$	
Exercise 4.7	$[6, 3, 4]_5$	
Exercise 4.8	$[7, 3, 5]_7$	
Exercise 4.9	$[9, 3, 6]_3$	
Exercise 4.14	$[10, 6, 4]_3$	
Example 5.5	$[11, 6, 5]_3$	Ternary Golay
Example 5.5	$[12, 6, 6]_3$	Extended ternary Golay
Example 5.9	$[23, 12, 7]_2$	Binary Golay
Example 5.9	$[24, 12, 8]_2$	Extended binary Golay
Example 5.11	$[31, 16, 7]_2$	
Example 5.12	$[n, n - d + 1, d]_q$	Cyclic Reed–Solomon
Exercise 5.6	$[15, 7, 5]_2$	
Exercise 5.6	$[31, 11, 11]_2$	
Exercise 5.6	$[13, 4, 7]_3$	
Exercise 5.7	$[17, 9, 5]_2$	Zetterberg
Exercise 5.7	$[18, 9, 6]_2$	Extended Zetterberg
Exercise 5.8	$[11, 6, 5]_4$	
Exercise 5.9	$[17, 9, 7]_4$	

(continued)

Example 6.4	$[q + 1, q - d + 2, d]_q$	Reed–Solomon
Exercise 6.9	$[2^h + 2, 3, 2^h]_{2^h}$	Translation hyperoval
Exercise 6.10	$[10, 5, 6]_9$	Glynn
Exercise 6.11	$[2^h + 1, 4, 2^h - 2]_{2^h}$	Segre
Example 7.2	$[10, 7, 3]_3$	
Example 7.14	$[8, 3, 5]_4$	
Example 7.14	$[8, 5, 3]_4$	
Exercise 7.10	$[24, 4, 18]_9$	
Example 8.4	$[12, 3, 6]_2$	Affine plane of order 3
Theorem 9.3	$[2^m, 1 + \binom{m}{1} + \dots + \binom{m}{r}, 2^{m-r}]_2$	Reed–Muller
Theorem 9.11	$(2^m, 2^{2m}, 2^{m-1} - 2^{m/2-1})_2$	Kerdock
Example 9.12	$(16, 256, 6)_2$	Nordstrom–Robinson
Exercise 10.6	$(12, 8, 6)_2$	
Exercise 10.7	$(16, 256, 6)_2$	Nordstrom–Robinson

Contents

	Table of Parameters for Codes in the Text	ix
1	Shannon's Theorem	1
1.1	Entropy	1
1.2	Information Channels	4
1.3	System Entropies and Mutual Information	5
1.4	Decoding and Transmission Rate	10
1.5	Shannon's Theorem	11
1.6	Comments	14
1.7	Exercises	14
2	Finite Fields	17
2.1	Definitions and Construction	17
2.2	Properties of Finite Fields	20
2.3	Factorisation of Cyclotomic Polynomials	21
2.4	Affine and Projective Spaces over Finite Fields	24
2.5	Comments	26
2.6	Exercises	26
3	Block Codes	29
3.1	Minimum Distance	29
3.2	Bounds on Block Codes	32
3.3	Asymptotically Good Codes	36
3.4	Comments	43
3.5	Exercises	43
4	Linear Codes	47
4.1	Preliminaries	47
4.2	Syndrome Decoding	51
4.3	Dual Code and the MacWilliams Identities	54
4.4	Linear Codes and Sets of Points in Projective Spaces	58
4.5	Griesmer Bound	59
4.6	Constructing Designs from Linear Codes	63
4.7	Comments	66
4.8	Exercises	67

5	Cyclic Codes	71
5.1	Basic Properties	71
5.2	Quadratic Residue Codes	75
5.3	BCH Codes	78
5.4	Comments	80
5.5	Exercises	81
6	Maximum Distance Separable Codes	83
6.1	Singleton Bound	84
6.2	Reed–Solomon Code	84
6.3	Linear MDS Codes	91
6.4	MDS Conjecture	94
6.5	Comments	100
6.6	Exercises	101
7	Alternant and Algebraic Geometric Codes	105
7.1	Subfield Subcodes	105
7.2	Generalised Reed–Solomon Codes	107
7.3	Alternant Codes Meeting the Gilbert–Varshamov Bound	109
7.4	Algebraic Geometric Codes	112
7.5	Algebraic Geometric Codes Surpassing the Gilbert–Varshamov Bound	117
7.6	Comments	119
7.7	Exercises	119
8	Low Density Parity Check Codes	123
8.1	Bipartite Graphs with the Expander Property	123
8.2	Low Density Parity Check Codes	126
8.3	Decoding LDPC Codes	128
8.4	Comments	131
8.5	Exercises	131
9	Reed–Muller and Kerdock Codes	133
9.1	Binary Reed–Muller Codes	133
9.2	Decoding Reed–Muller Codes	135
9.3	Kerdock Codes	142
9.4	Non-binary Reed–Muller Codes	145
9.5	Comments	148
9.6	Exercises	149

10	<i>p</i>-Adic Codes	151
10.1	<i>p</i> -Adic Numbers	151
10.2	Polynomials over the <i>p</i> -Adic Numbers	153
10.3	<i>p</i> -Adic Codes	155
10.4	Codes over $\mathbb{Z}/p^h\mathbb{Z}$	157
10.5	Codes over $\mathbb{Z}/4\mathbb{Z}$	160
10.6	Comments	162
10.7	Exercises	162
	Hints and Answers to Selected Exercises	165
	Bibliography	170
	Index	175



Shannon's Theorem

The content of this chapter is rather different in nature to what appears in the rest of the book, since Shannon's theorem is really a theorem from information theory and not coding theory. However, we include it here because it tells us that reliable communication can be achieved using a noisy channel and sets a limit for what is feasible in terms of the proportion of data we can send whilst being almost sure to be able to recover the original message from the distorted signal. Essentially, this chapter is a very brief introduction to information theory, the mathematics this entails is probabilistic in nature, whereas later it will be more algebraic and to some extent geometric. It is not essential to the rest of the text and can be treated as optional.

1.1 Entropy

Let $S = \{s_1, \dots, s_m\}$ be a finite set of values and suppose that we have a random variable X which takes the value $s_i \in S$ with a probability p_i . In other words, we have a probability function, where the probability that X is s_i is

$$P(X = s_i) = p_i.$$

Therefore, $0 \leq p_i \leq 1$, and

$$\sum_{i=1}^m p_i = 1.$$

This random variable may be the result of some kind of experiment, where S is the set of possible results, or a communication, where S is the set of possible symbols which can be sent (or received). Usually the value of a random variable is a real number (which allows us to calculate its expectation and other quantities dependent on the random variable) but we will not be so strict here. We will not calculate these quantities and will satisfy ourselves that a string of symbols is a legitimate value of the random variable.

Example 1.1

Let $S = \{2, 3, \dots, 12\}$ and let the probability that $X = s_i$ be given by

s_i	2	3	4	5	6	7	8	9	10	11	12
P_i	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

the probability that the sum of two dice throws is equal to s_i . ■

Let f be a function

$$f : (0, 1] \rightarrow [0, \infty),$$

with the property that

(I1) $f(x)$ is a continuous decreasing function and $f(x) = 0$ if $x = 1$,

(I2) $f(xy) = f(x) + f(y)$.

The function f can be interpreted as a measure of **information** we get if we consider f applied to a probability p , where p is the probability that a certain value is selected by a random variable X . The axiom (I1) indicates that more information is gained when a lower the probability event occurs. If we imagine that we are repeating the experiment, then (I2) is saying that the information we get from two elements is equal to the sum of the information we get from each element.

Lemma 1.2 *If f is a function satisfying (I1) and (I2), then*

$$f(x) = -\log_r(x),$$

for some $r > 1$.

Proof

Define $g(x) = f(e^{-x})$.

Then (I2) implies

$$g(x + y) = f(e^{-(x+y)}) = f(e^{-x}e^{-y}) = f(e^{-x}) + f(e^{-y}) = g(x) + g(y).$$

Therefore, g is a continuous additive function which implies that $g(x) = cx$ for some $c \in \mathbb{R}$.

Putting $y = e^{-x}$ gives

$$f(y) = g(-\ln y) = -c \ln y.$$

Since $\ln y$ is an increasing function of y and, according to (I1) f is a decreasing function, we have $c > 0$.

The lemma follows by letting $c = (\ln r)^{-1}$. □

To define a measure of the information we get from a random variable X , we weigh the sum of the information according to the probabilities.

The (r -ary) **entropy** $H_r(X)$ of X is

$$H_r(X) = \sum_{i=1}^m p_i f(p_i) = - \sum_{i=1}^m p_i \log_r(p_i).$$

We assume throughout that the function $x \log x$ is zero when evaluated at 0.

Example 1.3

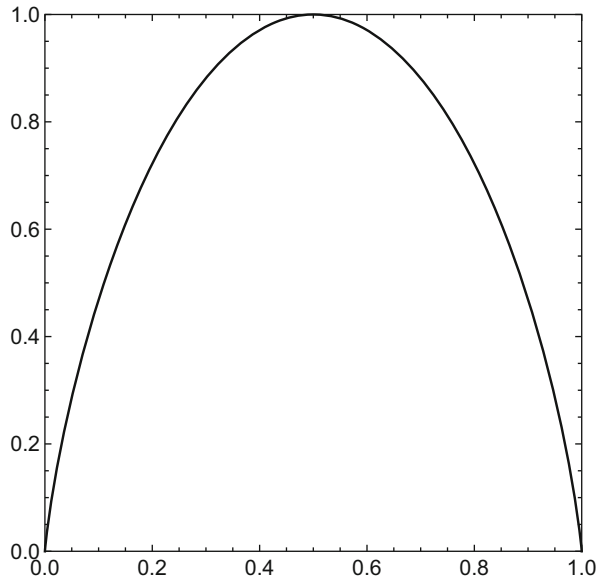
Suppose that $S = \{0, 1\}$ and that $P(X = 0) = p$ and $P(X = 1) = 1 - p$. Then, the binary entropy $H_2(X)$ of X is

$$h(p) = -p \log_2 p - (1 - p) \log_2(1 - p).$$

This function of p , defined on the real line interval $[0, 1]$, is called the **binary entropy function**. Its graph is drawn in [Figure 1.1](#). ■

Observe that for the entropy to be zero every term in the sum must be zero, since every term in the sum is non-negative. But $p_i \log_r(p_i) = 0$ implies p_i is either 0 or 1. Therefore, $P(X = s_i) = 1$, for some i and $P(X = s_j) = 0$ for all $j \neq i$. In this case X conveys no information, since there is no uncertainty. At the other end of the scale, intuitively, the most uncertainty, and so the most information, is conveyed when X is equally likely to be any of the elements of S . This is proven in the following theorem.

■ **Fig. 1.1** The binary entropy function on the interval $[0, 1]$.



Theorem 1.4

Let X be a random variable taking values from a finite set S . Then

$$H_r(X) \leq \log_r |S|$$

with equality if and only if $P(X = s) = 1/|S|$, for all $s \in S$.

Proof

Observe that $\ln x \leq x - 1$ with equality if and only if $x = 1$.

By definition,

$$\begin{aligned} H_r(X) - \log_r |S| &= - \sum_i p_i \log_r(p_i) - \sum_i p_i \log_r |S| \\ &= \sum_i p_i \log_r\left(\frac{1}{p_i |S|}\right) = \frac{1}{\ln r} \sum_i p_i \ln\left(\frac{1}{p_i |S|}\right) \leq \frac{1}{\ln r} \sum_i p_i \left(\frac{1}{p_i |S|} - 1\right) = 0. \end{aligned}$$

Equality occurs if and only if $p_i |S| = 1$ for all $i = 1, \dots, |S|$. □

1.2 Information Channels

Let X and Y be random variables which take values from finite sets S and T , respectively, according to the probability distributions $P(X = s_i) = p_i$ and $P(Y = t_j) = q_j$.

We consider the elements of S as the symbols which are sent and the elements of T as the symbols which are received. The channel through which the symbols are sent is denoted by Γ , which is defined by the matrix (p_{ij}) , where

$$p_{ij} = P(Y = t_j | X = s_i)$$

is the probability that t_j is received given that s_i was sent.

We define

$$q_{ij} = P(X = s_i | Y = t_j)$$

and

$$r_{ij} = P(X = s_i, Y = t_j).$$

The probabilities p_{ij} , q_{ij} and r_{ij} are called the **forwards probabilities**, the **backwards probabilities** and the **joint probabilities**, respectively.

Example 1.5

In a **binary symmetric channel** both $S = \{0, 1\}$ and $T = \{0, 1\}$. The channel is defined by the matrix

$$(p_{ij}) = \begin{pmatrix} \phi & 1 - \phi \\ 1 - \phi & \phi \end{pmatrix}$$

for some $\phi \in [0, 1]$. The rows and columns of the matrix are indexed by 0 and 1 in that order.

Suppose that X is the random variable defined by the probability

$$P(X = 0) = p.$$

From the channel matrix we can calculate the probabilities for Y from

$$P(Y = b) = \sum_{a \in \{0,1\}} P(Y = b | X = a)P(X = a).$$

For example,

$$P(Y = 0) = \phi p + (1 - p)(1 - \phi).$$

We can calculate the joint probabilities from

$$P(X = a, Y = b) = P(Y = b | X = a)P(X = a).$$

For example,

$$P(X = 1, Y = 0) = P(Y = 0 | X = 1)P(X = 1) = (1 - \phi)(1 - p).$$

We can calculate the backwards probabilities using

$$P(X = a | Y = b)P(Y = b) = P(X = a, Y = b).$$

For example,

$$P(X = 1 | Y = 0) = \frac{P(X = 1, Y = 0)}{P(Y = 0)} = \frac{(1 - \phi)(1 - p)}{\phi p + (1 - p)(1 - \phi)}.$$

■

1.3 System Entropies and Mutual Information

We define the **input entropy** and the **output entropy** as

$$H(X) = - \sum_i p_i \log p_i$$

and

$$H(Y) = - \sum_j q_j \log q_j,$$

respectively.

We suppress the r in the logarithm, but assume that it is the same for both definitions.

Given that we have received $t_j \in T$, we can calculate the entropy of X , conditional on the fact that we know $Y = t_j$, by using the backwards probabilities. This gives

$$H(X|Y = t_j) = - \sum_i q_{ij} \log q_{ij}.$$

This tells us the average information of X , knowing that $Y = t_j$.

If this is zero, then it would say that we know everything about X . This would mean that the backwards probabilities q_{ij} would be 1 for some i and zero for the others. In other words, that if we receive t_j , then we know which symbol was sent.

If this is $H(X)$, then this says that we learn nothing about X when we receive t_j . This would happen if

$$q_{ij} = P(X = s_i | Y = t_j) = P(X = s_i) = p_i.$$

Averaging over $t_j \in Y$, we obtain the **conditional entropy**, the average information of X knowing Y ,

$$H(X|Y) = - \sum_{i,j} q_j q_{ij} \log q_{ij}.$$

Similarly, the average information of Y knowing X is

$$H(Y|X) = - \sum_{i,j} p_j p_{ij} \log p_{ij}.$$

The **joint entropy** $H(X, Y)$ is given by the joint probabilities and is the average information gained from both the input and output,

$$H(X, Y) = - \sum_{i,j} r_{ij} \log r_{ij}.$$

Example 1.6

Suppose Γ is the channel with input random variable X , defined by

$$P(X = 0) = P(X = 1) = \frac{1}{2},$$

and output random variable Y taking a value from the set $\{0, 1, *\}$ and that the channel matrix

$$(p_{ij}) = \begin{pmatrix} \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

has rows indexed by 0 and 1 in that order and columns indexed by 0, 1, * in that order.

We can calculate directly

$$H(Y|X) = -\frac{1}{2} \left(\frac{3}{4} \log \frac{3}{4} + 2 \frac{1}{2} \log \frac{1}{2} + \frac{1}{4} \log \frac{1}{4} \right) = -\frac{3}{8} \log 3 + \frac{3}{2} \log 2$$

and

$$H(X) = -\frac{1}{2} (\log \frac{1}{2} + \log \frac{1}{2}) = \log 2.$$

We can calculate the output probabilities using $q_j = \sum_i p_i p_{ij}$. This gives

$$(q_0, q_1, q_*) = \left(\frac{3}{8}, \frac{1}{4}, \frac{3}{8} \right).$$

We can calculate the backwards probabilities q_{ij} using

$$q_{ij} = \frac{p_{ij} p_i}{q_j},$$

and obtain

$$(q_{ij}) = \begin{pmatrix} 1 & 0 & \frac{1}{3} \\ 0 & 1 & \frac{2}{3} \end{pmatrix}.$$

Therefore,

$$H(X|Y) = -\frac{3}{8} \left(\frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3} \right) = \frac{3}{8} \log 3 - \frac{1}{4} \log 2$$

and

$$H(Y) = -2 \frac{3}{8} \log \frac{3}{8} - \frac{1}{4} \log \frac{1}{4} = \frac{11}{4} \log 2 - \frac{3}{4} \log 3.$$

Observe that

$$H(Y) - H(Y|X) = -\frac{3}{8} \log 3 + \frac{5}{4} \log 2 = H(X) - H(X|Y).$$

Finally, we can calculate the joint probabilities from $r_{ij} = p_i p_{ij}$ and get

$$(r_{ij}) = \begin{pmatrix} \frac{3}{8} & 0 & \frac{1}{8} \\ 0 & \frac{1}{4} & \frac{1}{4} \end{pmatrix}.$$

The joint entropy is

$$H(X, Y) = -2\frac{1}{4} \log \frac{1}{4} - \frac{3}{8} \log \frac{3}{8} - \frac{1}{8} \log \frac{1}{8} = \frac{5}{2} \log 2 - \frac{3}{8} \log 3.$$

Observe that in [Example 1.6](#)

$$H(X, Y) = H(X|Y) + H(Y) = H(Y|X) + H(X).$$

This is no coincidence and holds in general.

Lemma 1.7 *For random variables X and Y defined on finite sets,*

$$H(X, Y) = H(X|Y) + H(Y) = H(Y|X) + H(X).$$

Proof

Since

$$P(X = s_i, Y = t_j) = P(X = s_i | Y = t_j)P(Y = t_j),$$

we have that

$$r_{ij} = q_j q_{ij}.$$

By direct calculation,

$$\begin{aligned} H(X, Y) &= - \sum_{i,j} r_{ij} \log r_{ij} = - \sum_{i,j} q_j q_{ij} \log q_j q_{ij} \\ &= - \sum_{i,j} q_j q_{ij} \log q_j - \sum_{i,j} q_j q_{ij} \log q_{ij}. \end{aligned}$$

Since $\sum_i q_{ij} = 1$,

$$H(X, Y) = - \sum_j q_j \log q_j + H(X|Y) = H(Y) + H(X|Y).$$

Reversing the roles of X and Y we obtain the second equality. □

The **mutual information** of X and Y is

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X),$$

the amount of information about X conveyed by Y and vice versa.

If $H(X) = H(X|Y)$, then Y tells us nothing about X , so the mutual information is zero. This is an unreliable channel and useless as a means of communication.

If $H(X|Y) = 0$, then knowing Y we know everything about X , so $I(X, Y) = H(X)$. This is the ideal situation, where when we receive something we know exactly what was sent.

Example 1.8

Suppose Γ is the channel with input random variable X defined on $\{0, 1, 2\}$ in which the probability $P(X = x) = \frac{1}{3}$, for all $x \in \{0, 1, 2\}$. Suppose Y is the output random variable defined on $\{0, 1\}$ and that the channel matrix is

$$(p_{ij}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix},$$

where the rows are indexed by 0, 1, 2 in that order and the columns are indexed by 0, 1 in that order.

The conditional entropy

$$H(Y|X) = 0.$$

This indicates that knowing X we know Y . Explicitly, we know that if 0 or 2 is sent, then 0 will be received, and if 1 is sent, then 1 will be received.

The entropy of X is

$$H(X) = -3 \frac{1}{3} \log \frac{1}{3} = \log 3.$$

The output probabilities, applying

$$q_j = \sum_i p_i p_{ij},$$

are

$$(q_0, q_1) = \left(\frac{2}{3}, \frac{1}{3}\right).$$

The backwards probabilities q_{ij} , applying

$$q_{ij} = \frac{p_{ij} p_i}{q_j},$$

are

$$(q_{ij}) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \\ \frac{1}{2} & 0 \end{pmatrix}.$$

Therefore,

$$H(X|Y) = -\frac{2}{3}2\left(\frac{1}{2} \log \frac{1}{2}\right) = \frac{2}{3} \log 2$$

and the entropy of Y is

$$H(Y) = -\frac{2}{3} \log \frac{2}{3} - \frac{1}{3} \log \frac{1}{3} = \log 3 - \frac{2}{3} \log 2.$$

The mutual information is

$$I(X, Y) = H(X) - H(X|Y) = \log 3 - \frac{2}{3} \log 2 = H(Y).$$

Observe that knowing Y , we do not know X , even though the reverse is true. ■

1.4 Decoding and Transmission Rate

Let Γ be a channel with input random variable X and output random variable Y defined on finite sets $S = \{s_1, \dots, s_m\}$ and $T = \{t_1, \dots, t_n\}$, respectively.

A **decoding** Δ is a map from $\{1, \dots, n\}$ to $\{1, \dots, m\}$, which induces a map from T to S by mapping t_j to $s_{\Delta(j)}$. This map we interpret as the receiver decoding t_j as $s_{\Delta(j)}$. For convenience sake, we will sometimes write $\Delta(v) = u$, where $v \in T$ and $u \in S$. The probability that a decoding is correct is

$$q_{\Delta(j)j} = P(X = s_{\Delta(j)}|Y = t_j),$$

the probability that $s_{\Delta(j)}$ was sent given that t_j was received.

The average probability P_{COR} of a correct decoding is

$$P_{\text{COR}} = \sum_j q_j q_{\Delta(j)j}.$$

In most applications we know (p_{ij}) , how the channel behaves, but not the probability distributions which define the random variables X and Y . We choose Δ to be the decoding map where $\Delta(j)$ is chosen so that $p_{\Delta(j)j} \geq p_{ij}$ for all i . This decoding map is called **maximum likelihood decoding**.

Suppose that X is defined on the elements of a set A with r symbols.

A **block code** C is a subset of A^n . We will often refer to a block code as simply a **code**.

The **(transmission) rate** of C is defined as

$$R = \frac{\log_r |C|}{n}.$$

Thus, $R = 1$ if and only if $|C| = r^n$ if and only if $C = A^n$.

The **capacity** of a channel Γ is

$$\Lambda = \max I(X, Y),$$

where we maximise over all input random variables X and output random variables Y . In other words, maximising over all probability distributions p_i and q_j .

Shannon's theorem tells us that given the channel, for sufficiently large n , there are block codes of A^n whose rate R is arbitrarily close to Λ for which, when we use maximum likelihood decoding, the probability P_{COR} is arbitrarily close to 1. To be able to prove Shannon's theorem we will require a few lemmas.

For any $u, v \in A^n$, the **Hamming distance** $d(u, v)$ between u and v is the number of coordinates in which they differ.

Lemma 1.9 *For the binary symmetric channel defined as in Example 1.5 and block code $C \subseteq \{0, 1\}^n$, maximum likelihood decoding is $\Delta(v) = u$, where u is the closest element of C to v with respect to the Hamming distance.*

Proof

Let ϕ denote the probability that a symbol does not change when sent through the binary symmetric channel.

Suppose that $d(u, v) = i$. Then

$$P(u \text{ was sent} \mid v \text{ was received}) = \phi^{n-i}(1-\phi)^i = \phi^n \left(\frac{1-\phi}{\phi}\right)^i,$$

which is a decreasing function of i (assuming $\phi > 1 - \phi$).

Then maximum likelihood decoding will give $\Delta(v) = u$, where u is the closest (with respect to the Hamming distance) n -tuple to v . \square

In general, the decoding map defined by $\Delta(v) = u$, where u is the closest (with respect to the Hamming distance) n -tuple to v is called **nearest neighbour decoding**.

1.5 Shannon's Theorem

We are almost in a position to prove Shannon's theorem for the binary symmetric channel. To be able to do so, we first calculate the channel capacity.

Lemma 1.10 *For the binary symmetric channel the capacity is*

$$\Lambda = 1 + \phi \log_2 \phi + (1 - \phi) \log_2(1 - \phi),$$

where ϕ denotes the probability that a symbol does not change.

Proof

Let p denote the probability that the input random variable X is 0. Let q denote the probability that the output random variable Y is 0. Then

$$H(Y) = -q \log_2 q - (1 - q) \log_2(1 - q).$$

The conditional entropy is

$$\begin{aligned} H(Y|X) &= - \sum_{i,j} p_i p_{ij} \log_2 p_{ij} \\ &= -p(\phi \log_2 \phi + (1 - \phi) \log_2(1 - \phi)) - (1 - p)(\phi \log_2 \phi + (1 - \phi) \log_2(1 - \phi)), \end{aligned}$$

and the mutual information is

$$I(X, Y) = -q \log_2 q - (1 - q) \log_2(1 - q) + \phi \log_2 \phi + (1 - \phi) \log_2(1 - \phi),$$

since

$$I(X, Y) = H(Y) - H(Y|X).$$

To obtain the channel capacity we maximise over all random variables X and Y , which in this case involves maximising over q . The function

$$h(q) = -q \log_2 q - (1 - q) \log_2(1 - q)$$

is maximised when $q = \frac{1}{2}$ (see [Figure 1.1](#)), where it has the value 1. □

Lemma 1.11 For $0 < \lambda \leq \frac{1}{2}$,

$$\sum_{i=0}^{\lfloor \lambda n \rfloor} \binom{n}{i} \leq 2^{nh(\lambda)}.$$

Proof

Observe that $\lambda/(1 - \lambda) \leq 1$ implies that, for $i = 1, \dots, \lfloor \lambda n \rfloor$,

$$\left(\frac{\lambda}{1 - \lambda}\right)^i \geq \left(\frac{\lambda}{1 - \lambda}\right)^{\lambda n}.$$

By the binomial theorem and the above inequality,

$$\begin{aligned} 1 &= (\lambda + 1 - \lambda)^n = \sum_{i=0}^n \binom{n}{i} \lambda^i (1 - \lambda)^{n-i} = \sum_{i=0}^n \binom{n}{i} \left(\frac{\lambda}{1 - \lambda}\right)^i (1 - \lambda)^n \\ &\geq \sum_{i=0}^{\lfloor \lambda n \rfloor} \binom{n}{i} \left(\frac{\lambda}{1 - \lambda}\right)^i (1 - \lambda)^n \geq \sum_{i=0}^{\lfloor \lambda n \rfloor} \binom{n}{i} \left(\frac{\lambda}{1 - \lambda}\right)^{\lambda n} (1 - \lambda)^n. \end{aligned}$$

Hence,

$$\sum_{i=0}^{\lfloor \lambda n \rfloor} \binom{n}{i} \leq \lambda^{-\lambda n} (1 - \lambda)^{-(1-\lambda)n}.$$

Since $\lambda^{-\lambda n} = 2^{-n\lambda \log_2 \lambda}$, the lemma follows. \square

The following theorem, [Theorem 1.12](#), is Shannon's noisy channel coding theorem.

Theorem 1.12

Let δ be an arbitrarily small positive real number and let R be a positive real number such that $R < \Lambda$. For all sufficiently large n , there is a code of length n and rate R , such that when we use maximum likelihood decoding the probability P_{COR} of a correct decoding is larger than $1 - \delta$.

Proof (for the binary symmetric channel)

Choose C to be a random subset of $\lfloor 2^{nR} \rfloor$ vectors of $\{0, 1\}^n$.

Let $u \in C$ and consider the transmission of u through the binary symmetric channel. On average, the number of coordinates which will change in the transmission of the n bits of u is $n(1 - \phi)$. As n gets large, the law of large numbers tells us that the probability that the number of symbols that change in the transmission varies from the average by a fixed constant tends to zero, so we can assume that the number of symbols which will change in the transmission of the n bits of u is $n(1 - \phi)$.

Suppose that we receive the n -tuple v . By [Lemma 1.9](#), we decode v to the n -tuple in C that is nearest to v with respect to the Hamming distance.

The probability that we mistakenly decode v to a different element of C , i.e., that there are other n -tuples of C with Hamming distance at most $n(1 - \phi)$ to v , is at most

$$\sum_{w \in C \setminus \{u\}} P(d(w, v) \leq n(1 - \phi)).$$

Counting the number of n -tuples at Hamming distance at most $(1 - \phi)n$ to v , and observing that there are 2^n n -tuples in total,

$$P(d(w, v) \leq n(1 - \phi)) < \frac{1}{2^n} \sum_{j=0}^{\lfloor (1-\phi)n \rfloor} \binom{n}{j}.$$

Since there are fewer than 2^{nR} n -tuples in $C \setminus \{u\}$ and these were chosen randomly,

$$1 - P_{\text{COR}} < 2^{nR} \frac{1}{2^n} \sum_{j=0}^{\lfloor (1-\phi)n \rfloor} \binom{n}{j}.$$

Lemma 1.11 implies that

$$1 - P_{\text{COR}} < 2^{n(R - (1 + \phi \log_2 \phi + (1 - \phi) \log_2(1 - \phi)))}.$$

By Lemma 1.10, the capacity of the binary symmetric channel is

$$\Lambda = 1 + \phi \log_2 \phi + (1 - \phi) \log_2(1 - \phi),$$

so

$$1 - P_{\text{COR}} < 2^{n(R - \Lambda)}.$$

Since $R < \Lambda$, for n sufficiently large,

$$2^{n(R - \Lambda)} < \delta,$$

which proves the theorem. \square

We have established that, given a channel with non-zero capacity, there are codes which allow us to communicate using the channel and decode with a probability of a correct decoding being close to 1. Our aim will be to find such codes which can be encoded and decoded in an efficient manner.

1.6 Comments

Although [Chapter 1](#) is primarily about Shannon's theorem [65], it is also a very brief introduction to information theory. For a more complete introduction, see the excellent book by Jones and Jones [40] or the classical introduction by Ash [3]. Jones and Jones [40] also contains an introduction to coding theory, albeit at a lower level to the treatment here. The idea to measure information dates back to Hartley's 1928 paper [35], although Shannon's work from the 1940s is widely acknowledged as the beginning of information theory. In 1961, Fano [23] proved that the capacity of a channel bounds the rate at which reliable transmissions can be achieved. For a proof of this, under some additional hypothesis, see Jones and Jones [40].

1.7 Exercises

1.1 Calculate the r -ary entropy $H_r(X)$ in [Example 1.1](#) and verify that $H_r(X) \leq \log_r 11$, as claimed by [Theorem 1.4](#).

1.2 Consider the random variable X defined on the elements of a set with n symbols where the probability that X is the i -th symbol is $\gamma/2^i$. Calculate the entropy $H_2(X)$ and verify that $H_2(X) \rightarrow 2$ as $n \rightarrow \infty$. Compare this with the bound $H_2(X) \leq \log_2 n$ given by [Theorem 1.4](#).

1.7 · Exercises

1.3 Let X be the random variable defined on a set S with three symbols with probabilities $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{4}$. Let X^n be the random variable defined on a set S^n in which the probability that $X^n = (s_1, \dots, s_n)$ is

$$\prod_{i=1}^n p_i,$$

where $P(X = s_i) = p_i$.

- Calculate, for how many symbols, X^n has probability 2^{j-2n} , where $j = 0, \dots, n$.
- Calculate $H_2(X)$ and $H_2(X^n)$ and verify that $H_2(X^n) = nH_2(X)$.

1.4 Calculate the mutual information for the **binary erasure channel** defined by the channel matrix

$$(p_{ij}) = \begin{pmatrix} \phi & 0 & 1 - \phi \\ 0 & \phi & 1 - \phi \end{pmatrix},$$

in terms of p and ϕ , where p is the probability that the input random variable is one of the symbols.

1.5 Let X and Y be random variables defined on the elements of a set S of size r , where $P(X = s) = 1/r$, for all $s \in S$, and define a channel by the matrix (p_{ij}) , where

$$p_{ii} = \phi, \quad \text{and} \quad p_{ij} = \frac{1 - \phi}{r - 1}, \quad i \neq j.$$

Calculate the entropy $H_r(Y)$ and the mutual information $I(X, Y)$.

1.6 Calculate P_{COR} for the binary symmetric channel, in which a bit changes with probability ϕ , using maximum likelihood decoding.

1.7 Prove that for the repetition code $C = \{00 \dots 0, 11 \dots 1\} \subset \{0, 1\}^n$, applying nearest neighbour decoding whilst transmitting through a reliable binary symmetric channel, $P_{\text{COR}} \rightarrow 1$ and $R \rightarrow 0$ as $n \rightarrow \infty$.

1.8 Let Γ be the channel with input random variable X taking values from $\{0, 1\}$, output random variable Y taking values from $\{0, 1, *\}$ and channel matrix

$$(p_{ij}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{4} & \frac{1}{4} \end{pmatrix}.$$

Let $p = P(X = 0)$.

- Calculate $H(Y|X)$, $H(Y)$ and $H(X)$ in terms of p .
- Calculate the capacity of the channel Γ .
- Give an interpretation of the fact that $H(X)$ equals the mutual information $I(X, Y)$.

1.9 Let Λ be the channel with channel matrix

$$(p_{ij}) = \begin{pmatrix} \frac{3}{4} & \frac{1}{8} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix},$$

where the input random variable X takes values from $\{0, 1\}$ and output random variable Y takes values from $\{0, 1, *\}$.

Let $p = P(X = 0)$.

- i. Calculate the entropy $H(Y)$ as a function of p .
- ii. Prove that the mutual information is

$$I(X, Y) = -\frac{3}{4}p \log p - \frac{1}{4}(4 - 3p) \log(4 - 3p) + (2 - 2p) \log 2.$$

- iii. Calculate the channel capacity of Λ .



Finite Fields

This chapter is a brief introduction to finite fields. The most important facts that will be established are that finite fields necessarily contain p^h elements, for some prime number p and positive integer h , and that the field with p^h elements is unique, up to isomorphism. We will study how to factorise cyclotomic polynomials over finite fields, which is used in ► Chapter 5 to construct cyclic codes. The projective and affine space over a finite field are also introduced and will appear later in various chapters.

2.1 Definitions and Construction

A **commutative ring** R is a set with two binary operations, addition and multiplication, such that it is a commutative group with respect to addition with identity element 0, and multiplication is commutative ($ab = ba$), associative ($(ab)c = a(bc)$), distributive ($a(b + c) = ab + ac$) and has an identity element 1.

A **field** is a commutative ring in which every non-zero element has a multiplicative inverse. In other words, for all $a \neq 0$, there is a b such that $ab = 1$. In particular, this implies that if $ab = 0$, then either $a = 0$ or $b = 0$.

Example 2.1

The rational numbers \mathbb{Q} , the real numbers \mathbb{R} and the complex numbers \mathbb{C} are all fields. ■

In the above example, the sum $1 + \dots + 1$ is never zero.

Let \mathbb{F} be a field with multiplicative identity 1. Suppose there is a n for which summing n ones gives zero and let n be minimal with this property. If p is a divisor of n , then

$$\underbrace{1 + \dots + 1}_n = \underbrace{(1 + \dots + 1)}_p \underbrace{(1 + \dots + 1)}_{n/p},$$

which contradicts the minimality of n , since the left-hand is zero implies one of the terms in the product on the right-hand side is zero. It follows that n is a prime p . The number

p is called the **characteristic** of the field. If no such n exists, then the characteristic is zero.

Throughout the remainder of this chapter p will be a prime number.

Example 2.2

The ring $\mathbb{Z}/p\mathbb{Z}$, the integers modulo p , is a field of characteristic p and is denoted \mathbb{F}_p . ■

The ring $\mathbb{Z}/n\mathbb{Z}$ is not a field when n is not a prime, since it has non-zero elements which have no multiplicative inverse.

In the following theorem, (f) denotes the set of elements of the ring of polynomials $\mathbb{F}_p[X]$ which are multiples of the polynomial f . The elements of the quotient ring $\mathbb{F}_p[X]/(f)$ are cosets of the form $g + (f)$, where addition is defined as

$$g + (f) + h + (f) = g + h + (f)$$

and multiplication is defined as

$$(g + (f))(h + (f)) = gh + (f).$$

One can think of the quotient ring as the polynomials modulo f .

Theorem 2.3

If f is an irreducible polynomial in the ring $\mathbb{F}_p[X]$, then $\mathbb{F}_p[X]/(f)$ is a field of characteristic p .

Proof

We have to show that $g + (f)$ has a multiplicative inverse for all $g \in \mathbb{F}_p[X]$, such that $g + (f) \neq 0 + (f)$.

Let

$$\mathcal{B} = \{gh + (f) \mid h + (f) \in \mathbb{F}_p[X]/(f)\}.$$

If

$$gh_1 + (f) = gh_2 + (f)$$

then

$$g(h_1 - h_2) + (f) = 0 + (f).$$

Since f is irreducible and g is not a multiple of f , this implies that $h_1 - h_2$ is a multiple of f , which implies

$$h_1 + (f) = h_2 + (f).$$

Therefore, if

$$h_1 + (f) \neq h_2 + (f)$$

then

$$gh_1 + (f) \neq gh_2 + (f)$$

In particular, there is an element $h + (f)$ for which and so \mathcal{B} contains as many elements as the finite set $\mathbb{F}_p[X]/(f)$.

$$(g + (f))(h + (f)) = 1 + (f),$$

so $g + (f)$ has a multiplicative inverse. □

If f is an irreducible polynomial of degree h , then $\mathbb{F}_p[X]/(f)$ is a field with p^h elements.

For example, [Table 2.1](#) is the addition and multiplication table of

$$\mathbb{F}_2[X]/(X^2 + X + 1),$$

a finite field with four elements and [Table 2.2](#) is the multiplication table of

$$\mathbb{F}_3[X]/(X^2 + 1),$$

a finite field with nine elements.

Table 2.1 The addition and multiplication table for the field $\mathbb{F}_2[X]/(X^2 + X + 1)$.

+	0	1	X	1+X	.	0	1	X	1+X
0	0	1	X	1+X	0	0	0	0	0
1	1	0	1+X	X	1	0	1	X	1+X
X	X	1+X	0	1	X	0	X	1+X	1
1+X	1+X	X	1	0	1+X	0	1+X	1	X

Table 2.2 The multiplication table for the field $\mathbb{F}_3[X]/(X^2 + 1)$.

.	0	1	2	X	1+X	2+X	2X	1+2X	2+2X
0	0	0	0	0	0	0	0	0	0
1	0	1	2	X	1+X	2+X	2X	1+2X	2+2X
2	0	2	1	2X	2+2X	1+2X	X	2+X	1+X
X	0	X	2X	2	2+X	2+2X	1	1+X	1+2X
1+X	0	1+X	2+2X	2+X	2X	1	1+2X	2	X
2+X	0	2+X	1+2X	2+2X	1	X	1+X	2X	2
2X	0	2X	X	1	1+2X	1+X	2	2+2X	2+X
1+2X	0	1+2X	2+X	1+X	2	2X	2+2X	X	1
2+2X	0	2+2X	1+X	1+2X	X	2	2+X	1	2X

2.2 Properties of Finite Fields

Throughout the remainder of this chapter q will be a power of the prime number p .

Theorem 2.4

For all x in a finite field \mathbb{F} with q elements $x^q = x$.

Proof

Suppose $x \neq 0$. Let $\mathcal{A} = \{ax \mid a \in \mathbb{F}\}$. Since $a_1x \neq a_2x$, for $a_1 \neq a_2$, the set \mathcal{A} consists of all the elements of \mathbb{F} . If we multiply together the non-zero elements of \mathcal{A} , then we obtain the product of all the non-zero elements of \mathbb{F} ,

$$\prod_{a \in \mathbb{F} \setminus \{0\}} ax = \prod_{a \in \mathbb{F} \setminus \{0\}} a,$$

which implies $x^{q-1} = 1$. □

Let \mathbb{F}_q denote the splitting field of the polynomial $X^q - X$ over the field \mathbb{F}_p , that is the smallest field extension of \mathbb{F}_p in which $X^q - X$ factorises into linear factors.

Theorem 2.5

A finite field with q elements is isomorphic to \mathbb{F}_q .

Proof

By [Exercise 2.1](#), $q = p^h$ for some prime p . The theorem follows from the uniqueness of splitting fields and [Theorem 2.4](#). □

In practice, when we work over a field with q elements, we fix an irreducible polynomial f of degree h and compute in the ring $\mathbb{F}_p[X]/(f)$ which, by [Theorem 2.3](#), is a field with p^h elements.

Lemma 2.6 *The map σ which maps $x \mapsto x^p$ is an automorphism of \mathbb{F}_q .*

Proof

The characteristic of \mathbb{F}_q is p , so

$$(x + y)^p = \sum_{j=0}^p \binom{p}{j} x^j y^{p-j} = x^p + y^p,$$

for all $x, y \in \mathbb{F}_q$.

Note that the binomial coefficient is an element of the field \mathbb{F}_p and since $\binom{p}{j}$ is divisible by p for $j = 1, \dots, p-1$, it is zero.

Clearly $(xy)^p = x^p y^p$, so σ preserves the additive and multiplicative structure of the field. \square

The automorphism σ in [Lemma 2.6](#) is called the **Frobenius automorphism**. Observe that the group of automorphisms generated by σ is a cyclic group of order h , where $q = p^h$.

2.3 Factorisation of Cyclotomic Polynomials

In this section we will see how cyclotomic polynomials factorise over finite fields. These factorisations will be used principally in [Chapter 5](#) and implicitly in [Chapter 10](#).

As in the previous section $q = p^h$, for some prime p .

By [Lemma 2.4](#), the polynomial $X^{q-1} - 1$ factorises into distinct linear factors in $\mathbb{F}_q[X]$.

Lemma 2.7 *The polynomial $X^{q-1} - 1$ factorises in $\mathbb{F}_p[X]$ into distinct irreducible factors whose degrees are divisors of h .*

Proof

Let $\epsilon \in \mathbb{F}_q$ and let $\mathbb{F}_p(\epsilon)$ denote the smallest extension field of \mathbb{F}_p containing ϵ .

Since $\epsilon \in \mathbb{F}_q$, $\mathbb{F}_p(\epsilon)$ is a subfield of \mathbb{F}_q . This subfield is generated as a vector space over \mathbb{F}_p by $1, \epsilon, \dots, \epsilon^{r-1}$, where r is the dimension of this vector space. Hence, there are $a_0, \dots, a_r \in \mathbb{F}_p$, such that

$$a_0 + a_1\epsilon + \dots + a_r\epsilon^r = 0.$$

This implies ϵ is a zero of the polynomial

$$a_0 + a_1X + \dots + a_rX^r.$$

For the elements in $\mathbb{F}_p(\epsilon)$, [Lemma 2.4](#) implies $x^{p^r} = x$. Since $x^{p^h} = x$, this implies $X^{p^r} - X$ divides $X^{p^h} - X$ which, by [Exercise 2.3](#), implies r divides h . \square

Observe that in the following example $X^8 - X = X^8 + X$, since $p = 2$.

Example 2.8

The polynomial $X^8 + X$ factorises in $\mathbb{F}_2[X]$ as

$$(X^3 + X + 1)(X^3 + X^2 + 1)(X + 1)X.$$

Suppose that e is a root of $X^3 + X + 1$. Then $e^3 = e + 1$, $e^4 = e^2 + e$, $e^5 = e^2 + e + 1$, $e^6 = e^2 + 1$ and $e^7 = 1$. Therefore, every non-zero element of $\mathbb{F}_2(e) \cong \mathbb{F}_8$ is a power of e . ■

An element e with the property that every non-zero element $x \in \mathbb{F}_q$ can be written as $x = e^i$, for some i , is called **primitive**.

Example 2.9

The polynomial $X^9 - X$ factorises in $\mathbb{F}_3[X]$ as

$$(X^2 + X - 1)(X^2 - X - 1)(X^2 + 1)(X - 1)(X + 1)X.$$

■

Observe that the elements of \mathbb{F}_9 which are roots of $X^2 + 1$ are not primitive.

Example 2.10

The polynomial $X^{q-1} - 1$ factorises as

$$(X^{(q-1)/2} - 1)(X^{(q-1)/2} + 1)$$

when q is odd. The roots of the first polynomial are the non-zero squares in \mathbb{F}_q and the roots of the second factor are the non-squares in \mathbb{F}_q . Note that the squares are never primitive elements. If -1 is a square, then $-1 = a^2$, for some $a \in \mathbb{F}_q$, which implies $(-1)^{(q-1)/2} = a^{q-1} = 1$ and so $q \equiv 1$ modulo 4. ■

Lemma 2.11 *The polynomial*

$$(X - \alpha_1) \cdots (X - \alpha_m) \in \mathbb{F}_q[X]$$

if and only if

$$\{\alpha_1, \dots, \alpha_m\} = \{\alpha_1^q, \dots, \alpha_m^q\}$$

as multi-sets.

Proof

Let

$$f(X) = (X - \alpha_1) \cdots (X - \alpha_m) = a_0 + a_1X + \cdots + a_mX^m.$$

2.3 · Factorisation of Cyclotomic Polynomials

Since

$$(X - \alpha_1^q) \cdots (X - \alpha_m^q) = a_0^q + a_1^q X + \cdots + a_m^q X^m,$$

$f \in \mathbb{F}_q[X]$ if and only if

$$f(X) = (X - \alpha_1^q) \cdots (X - \alpha_m^q).$$

□

Suppose we want to factorise $X^n - 1$ in $\mathbb{F}_q[X]$.

Our first observation is that if $(n, q) = q' > 1$, then

$$X^n - 1 = (X^{n/q'} - 1)^{q'},$$

since

$$\binom{q'}{j} = 0,$$

for $j = 1, \dots, q' - 1$.

Hence, it suffices to know how to factorise $X^m - 1$, where $(m, q) = 1$, in order to be able to factorise $X^n - 1$.

We look for an extension field of \mathbb{F}_q which contains n -th roots of unity by finding the minimum h such that n divides $q^h - 1$. This is equivalent to calculating the (multiplicative) order of q in $\mathbb{Z}/n\mathbb{Z}$.

An element $\epsilon \in \mathbb{F}_{q^h}$ is a **primitive n -th root of unity** if $\{1, \epsilon, \dots, \epsilon^{n-1}\}$ is the set of all n -th roots of unity.

Lemma 2.12 *Suppose $(n, q) = 1$ and let $\epsilon \in \mathbb{F}_{q^h}$ be a primitive n -th root of unity. The irreducible factors of $X^n - 1$ in $\mathbb{F}_q[X]$ are given by polynomials*

$$(X - \epsilon^r)(X - \epsilon^{rq}) \cdots (X - \epsilon^{rq^{d-1}}), \quad (2.1)$$

for $r = 0, \dots, n-1$, where d , which depends on r , is the minimum positive integer such that $rq^d \equiv r$ modulo n .

Proof

By [Lemma 2.11](#),

$$g(X) = (X - \epsilon^r)(X - \epsilon^{rq}) \cdots (X - \epsilon^{rq^{d-1}}) \in \mathbb{F}_q[X].$$

Suppose that

$$f = \sum_{i=0}^m a_i X^i \in \mathbb{F}_q[X]$$

and that $f(\alpha) = 0$. Then

$$0 = \sum_{i=0}^m a_i \alpha^i = \sum_{i=0}^m a_i \alpha^{iq},$$

which implies $f(\alpha^q) = 0$.

Therefore, $g(X)$ is the minimal degree polynomial in $\mathbb{F}_q[X]$ which is zero at α^r .

Hence, g is irreducible in $\mathbb{F}_q[X]$. \square

For each $r \in \{0, \dots, n-1\}$, the set

$$\{r, rq, rq^2, \dots, rq^{d-1}\},$$

where the elements of the set are computed modulo n , is called a **cyclotomic coset**.

The set $\{0, \dots, n-1\}$ splits into the disjoint union of cyclotomic cosets. Considering the polynomial in (2.1), we see that a cyclotomic coset of size d corresponds to an irreducible factor of degree d of $X^n - 1$ in its factorisation over \mathbb{F}_q .

Example 2.13

Suppose we wish to factorise $X^{12} - 1$ in $\mathbb{F}_{17}[X]$.

Since $17 \equiv 5 \pmod{12}$, the cyclotomic cosets are

$$\{0\}, \{1, 5\}, \{2, 10\}, \{3\}, \{4, 8\}, \{6\}, \{7, 11\}, \{9\}.$$

By Lemma 2.12, there are four factors in $\mathbb{F}_{17}[X]$ of degree 2 and four factors of degree one.

Alternatively, and somewhat intuitively, $X^{12} - 1$ factorises as $(X^6 - 1)(X^6 + 1)$ and

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

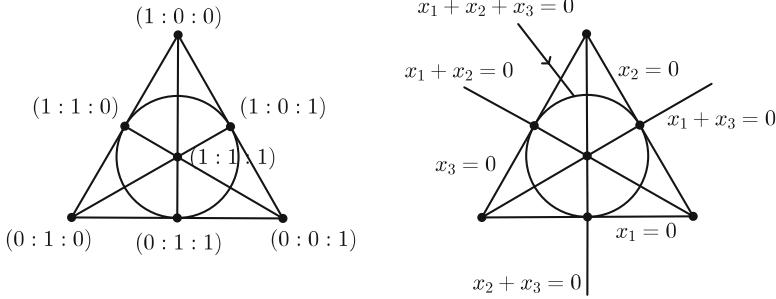
Moreover, $-((X/4)^6 - 1) \equiv X^6 + 1 \pmod{17}$, so

$$\begin{aligned} X^6 + 1 &\equiv -((X/4 - 1)((X/4)^2 + (X/4) + 1)((X/4) + 1)((X/4)^2 - (X/4) + 1)) \\ &\equiv (X - 4)(X^2 + 4X - 1)(X + 4)(X^2 - 4X - 1) \pmod{17}, \end{aligned}$$

which gives an explicit factorisation of $X^{12} - 1$. \blacksquare

2.4 Affine and Projective Spaces over Finite Fields

The **projective space** $\text{PG}(k-1, q)$ is the geometry whose i -dimensional subspaces are the $(i+1)$ -dimensional subspaces of the vector space \mathbb{F}_q^k , for $i = 0, \dots, k-2$. The 0, 1, 2-dimensional subspaces of $\text{PG}(k-1, q)$ are called **points**, **lines**, **planes**, respectively. The dimension shift is necessary so that familiar geometric properties hold, such as two points being joined by a line or three non-collinear points span a plane. A **hyperplane**



■ **Fig. 2.1** The projective plane over the field of two elements.

is a subspace of co-dimension 1, that is a subspace of one dimension less than the whole space. A hyperplane of a projective space can be defined as the kernel of a linear form (i.e. the set of zeros of a linear form). We use the notation

$$(x_1 : x_2 : \dots : x_k)$$

to denote the point of the projective space $\text{PG}(k - 1, q)$ which corresponds to the one-dimensional subspace spanned by the vector (x_1, x_2, \dots, x_k) of \mathbb{F}_q^k .

The geometry of points and lines drawn in ■ **Figure 2.1** is $\text{PG}(2, 2)$. In the left-hand copy the points have been labelled and in the right-hand copy the lines have been labelled.

The relevance of projective geometries in the study of error-correcting codes is partly due to the following. Suppose that the rows of a $k \times n$ matrix G form a basis of a k -dimensional subspace of \mathbb{F}_q^n . A vector of the subspace is

$$(u_1, \dots, u_n) = (a_1, \dots, a_k)G$$

for some $a = (a_1, \dots, a_k) \in \mathbb{F}_q^k$.

We will be particularly interested in how many zero coordinates the vector u has. Let s_i be the i -th column of the matrix G . We suppose that $s_i \neq 0$, for all $i = 1, \dots, n$.

Let \cdot denote the standard scalar product defined on \mathbb{F}_q^k .

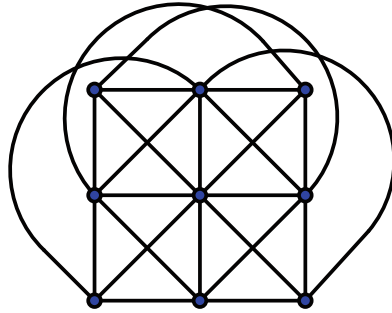
Observe that $u_i = 0$ if and only if

$$a \cdot s_i = 0$$

if and only if

$$\mu a \cdot \lambda s_i = 0$$

for any non-zero $\lambda, \mu \in \mathbb{F}_q$. Consider the columns of G as a set (or possibly multi-set) of points in the projective space $\text{PG}(k - 1, q)$.



■ **Fig. 2.2** The affine plane over the field of three elements.

Let π_a be the hyperplane which is the kernel of the linear form

$$a_1 X_1 + \cdots + a_k X_k.$$

In geometric terms, the above says that $u_i = 0$ if and only if the point s_i is incident with the hyperplane π_a . We will return to this in ► [Section 4.4](#).

The **affine space** $AG(k, q)$ is the geometry whose i -dimensional subspaces are the cosets of the i -dimensional subspaces of the vector space \mathbb{F}_q^k , $i = 0, \dots, k - 1$. The geometry of points and lines drawn in ■ [Figure 2.2](#) is $AG(2, 3)$. As in the projective space, the 0, 1, 2 and $(k - 1)$ -dimensional subspaces of $AG(k, q)$ are called **points**, **lines**, **planes** and **hyperplanes**, respectively.

2.5 Comments

The standard reference for finite fields is Lidl and Neiderreiter [46], a comprehensive text dedicated to finite fields. The more recent [52] contains a wealth of results concerning finite fields and their applications. There is a chapter on finite geometries by Cameron in [16] and Ball [6] is a textbook dedicated to the subject.

2.6 Exercises

- 2.1 Prove that a finite field has p^h elements, for some prime p and positive integer h .
- 2.2 Construct the addition and multiplication table for a field $\mathbb{F}_3[X]/(X^2 + X + 2)$.
- 2.3 Prove that $X^{p^r} - X$ divides $X^{p^h} - X$ if and only if r divides h . Conclude that a finite field with p^h elements has a subfield with p^r elements if and only if r divides h .
- 2.4 Use one of the irreducible factors of degree 3 in the factorisation of $X^7 - 1$ in $\mathbb{F}_2[X]$ to construct the multiplication table for a field with eight elements.

2.5 Determine the degrees of the irreducible factors of

- i. $X^{15} - 1$ in $\mathbb{F}_{17}[X]$,
- ii. $X^{23} - 1$ in $\mathbb{F}_2[X]$,
- ii. $X^{12} - 1$ in $\mathbb{F}_5[X]$.

2.6


- i. Factorise $X^9 - 1$ in $\mathbb{F}_7[X]$.
- ii. Factorise $X^{11} - 1$ in $\mathbb{F}_{19}[X]$.
- iii. Factorise $X^8 - 1$ in $\mathbb{F}_{19}[X]$.


2.7 Suppose n is prime and q is primitive in \mathbb{F}_n . Prove that $X^n - 1$ factorises into irreducible factors in $\mathbb{F}_q[X]$ as

$$(X - 1)(X^{n-1} + \cdots + X + 1).$$

2.8 Prove that

$$\sum_{x \in \mathbb{F}_q} x^j = \begin{cases} 0 & \text{if } q - 1 \text{ does not divide } j, \\ -1 & \text{if } q - 1 \text{ does divide } j \neq 0. \end{cases}$$

2.9 Label the points of  Figure 2.2 with the vectors from \mathbb{F}_3^2 (which are the cosets of the zero-vector) in such a way that three points are collinear if and only if the three vectors are contained in a coset of a one-dimensional subspace of the vector space. Note that a coset of a one-dimensional subspace of the vector space contains the vectors (x, y) which are the zeros of an equation $y = mx + c$ for some $m, c \in \mathbb{F}_3$ or an equation $x = c$, for some $c \in \mathbb{F}_3$.

2.10 By adding four points and one line to  Figure 2.2 and extending the lines of the affine plane with one point each, complete the affine plane $\text{AG}(2, 3)$ to the projective plane $\text{PG}(2, 3)$.

2.11

- i. Prove that the number of ordered r -tuples of r linearly independent vectors in \mathbb{F}_q^k is

$$(q^k - 1)(q^k - q) \cdots (q^k - q^{r-1}).$$

- ii. Prove that the number of $(r - 1)$ -dimensional subspaces in $\text{PG}(k - 1, q)$ is

$$\frac{(q^k - 1)(q^{k-1} - 1) \cdots (q^{k-r+1} - 1)}{(q^r - 1)(q^{r-1} - 1) \cdots (q - 1)}.$$

- iii. Prove that the number of $(r - 1)$ -dimensional subspaces in $\text{PG}(k - 1, q)$ containing a fixed $(s - 1)$ -dimensional subspace is equal to the number of $(r - s - 1)$ -dimensional subspaces in $\text{PG}(k - s - 1, q)$

2.12 Prove that the geometry one obtains by deleting a hyperplane from the projective space $\text{PG}(k, q)$ is isomorphic to the affine space $\text{AG}(k, q)$.



Block Codes

The main parameters of an error correcting block code, which we will often refer to simply as a code, are its length and minimum distance. In this chapter, we shall primarily be concerned with the relationship between the size of the code and these parameters. If we fix the length of the code, then we wish to maximise the minimum distance and the size of the code, which are contrary aims. If we fix the minimum distance too, then we simply consider the problem of maximising the size of the code. We shall prove the Gilbert–Varshamov lower bound, which is obtained by constructing block codes of a given length and minimum distance by applying the greedy algorithm. We will prove various upper bounds which will put limits on just how good a block code one can hope to find of a fixed length and minimum distance. Since Shannon’s theorem is an asymptotic result telling us what rates we can achieve with a code of arbitrarily long length, we shall for a large part of this chapter focus on sequences of codes whose length tends to infinity. If we use nearest neighbour decoding then, so that the probability we decode correctly does not tend to zero, we will be interested in finding sequences of codes for which both the transmission rate and the ratio of the minimum distance to the length are bounded away from zero. We set aside trying to answer the question of how these codes are implemented until later chapters in which we work with codes which have more structure.

3.1 Minimum Distance

Let A be a finite set of r elements called the **alphabet**. Recall that a **block code** (or simply a **code**) C is a subset of A^n . We say that C is an **r -ary code of length n** . A 2-ary code is called a **binary code** and a 3-ary code is called a **ternary code**.

Our aim will be to choose C so that the **codewords**, the elements of C , are far apart with respect to the Hamming distance. In this way, the code will have good error-correcting properties when we use nearest neighbour decoding.

Lemma 3.1 Let $u, v, w \in A^n$. The Hamming distance satisfies the triangle inequality

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

Proof

If u and v differ in the i -th coordinate, then w differs from one of u or v in the i -th coordinate. \square

Since we will use no other metric on A^n , we will refer to the Hamming distance between two elements u and v of A^n as the **distance** between u and v .

The **minimum distance** d of a code C is the minimum distance between any two codewords of C .

Lemma 3.2 Using nearest neighbour decoding, a block code of minimum distance d can correct up to $\frac{1}{2}(d - 1)$ errors.

Proof

For any $w \in A^n$ and codewords u and v , Lemma 3.1 implies

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

Hence, there is at most one codeword at distance at most $\frac{1}{2}(d - 1)$ from w . \square

Example 3.3

Let $A = \{a_1, \dots, a_r\}$. The r -ary repetition code C of length n is a block code with r codewords where, for each $a \in A$, there is a codeword in which a is repeated n times. The minimum distance of C is n , so C can correct up to $\frac{1}{2}(n - 1)$ errors using nearest neighbour decoding. The transmission rate of C is $\log_r |C|/n = 1/n$.

bit	a_1	a_2	\dots	a_r
codeword	$a_1 \cdots a_1$	$a_2 \cdots a_2$	\dots	$a_r \cdots a_r$



Example 3.4

The code

$$C = \{000000, 001011, 010101, 100110, 011110, 101101, 110011, 111000\}$$

is a block code of length 6 with 8 codewords which can correct up to 1 error, since it has minimum distance 3. We can assign to each codeword a distinct triple, which corresponds to the first three bits of the codeword. In this way, for each 3-tuple of bits, we send 6 bits. This is coherent with the definition of the transmission rate of C , which is $(\log_2 8)/6 = 1/2$.

3.1 · Minimum Distance

triple	000	001	010	100	011	101	110	111
codeword	000000	001011	010101	100110	011110	101101	110011	111000

■

Let C be a block code of length n and minimum distance d . An **extension** of C is code \overline{C} of length $n + 1$ obtained by adding a coordinate to each codeword of C in such a way that \overline{C} has minimum distance $d + 1$. The code \overline{C} is called an **extended code** of the code C .

Theorem 3.5

If C is a binary code of length n and minimum distance d and d is odd, then C has an extension.

Proof

We can suppose that $C \subset \{0, 1\}^n$. Let

$$\overline{C} = \{(u_1, \dots, u_n, u_{n+1}) \mid (u_1, \dots, u_n) \in C\},$$

where

$$u_{n+1} = u_1 + u_2 + \dots + u_n \pmod{2}.$$

Suppose that u and v are two codewords of C .

If $d(u, v) \geq d + 1$, then their corresponding codewords are at distance at least $d + 1$ in \overline{C} too.

Suppose $d(u, v) = d$. Consider the sum of the coordinates of u and v modulo 2. In $n - d$ coordinates u and v are the same, so these $n - d$ coordinates contribute zero to the sum. In d coordinates they are different, so we sum d ones and d zeros, which gives d . Since d is odd, the sum is non-zero modulo 2. Therefore, $u_{n+1} \neq v_{n+1}$ and the distance between u and v in \overline{C} is $d + 1$. □

Example 3.6

Let C be the binary block code of length 6 and minimum distance 3 from [Example 3.4](#). The extended code \overline{C} is a binary block code of length 7 and minimum distance 4.

codeword in C	000000	001011	010101	100110
codeword in \overline{C}	0000000	0010111	0101011	1001101
codeword in C	011110	101101	110011	111000
codeword in \overline{C}	0111100	1011010	1100110	1110001

3.2 Bounds on Block Codes

Let $A_r(n, d)$ denote the maximum $|C|$, for which there exists a r -ary block code C of length n and minimum distance d .

Theorem 3.7 (Gilbert–Varshamov bound)

We have the lower bound

$$A_r(n, d) \left(1 + \binom{n}{1}(r-1) + \cdots + \binom{n}{d-1}(r-1)^{d-1} \right) \geq r^n.$$

■

Proof

Let A be the alphabet, a set of r elements such that $C \subseteq A^n$, and suppose that C is a block code of size $A_r(n, d)$.

Let $u \in C$.

The set $B_{d-1}(u)$, of n -tuples in A^n at distance at most $d-1$ to u , has size

$$1 + \binom{n}{1}(r-1) + \cdots + \binom{n}{d-1}(r-1)^{d-1},$$

since there are precisely $\binom{n}{j}(r-1)^j$ of the n -tuples of A^n at distance j to u .

If

$$|C| \left(1 + \binom{n}{1}(r-1) + \cdots + \binom{n}{d-1}(r-1)^{d-1} \right) < r^n,$$

then there is an n -tuple which is at distance at least d from all the codewords of C . Therefore, C is not a code of length n and minimum distance d of maximum size, a contradiction. ■

Recall that the binary entropy function was defined as

$$h(p) = -p \log_2(p) - (1-p) \log_2(1-p).$$

For a code of length n and minimum distance d , we define the **relative minimum distance** to be $\delta = d/n$.

Corollary 3.8 *The following inequality holds*

$$\frac{1}{n} \log A_2(n, d) \geq 1 - h(\delta).$$

Proof

By [Lemma 1.11](#) and [Theorem 3.7](#),

$$A_2(n, d)2^{nh(\delta)} \geq 2^n.$$

Take logarithms of both sides and divide by n . ■

The sphere-packing problem in \mathbb{R}^n asks how many spheres can we pack into a box of given dimensions or, equivalently, what is the maximum ratio of spheres to the volume of the box one can achieve. In three dimensions, one can think of packing oranges into a box and trying to maximise the percentage of space inside the box which is taken up with oranges. In the discrete world, the analogous problem of packing spheres gives us the following theorem.

Let $t = \lfloor (d-1)/2 \rfloor$. In deference to [Lemma 3.2](#), we will sometimes call a code with minimum distance d , a t -error correcting code.

Theorem 3.9 (Sphere packing bound)

We have the upper bound

$$A_r(n, d) \left(1 + \binom{n}{1}(r-1) + \dots + \binom{n}{t}(r-1)^t \right) \leq r^n.$$

Proof

Let A be the alphabet of size r .

Let $C \subseteq A^n$ be a code of size $A_r(n, d)$.

Suppose that $u \in C$.

The set $B_t(u)$, of n -tuples in A^n at distance at most t to u , has size

$$|B_t(u)| = 1 + \binom{n}{1}(r-1) + \dots + \binom{n}{t}(r-1)^t.$$

For any pair $u, v \in C$, suppose that $w \in B_t(u) \cap B_t(v)$. By [Lemma 3.1](#),

$$d(u, v) \leq d(u, w) + d(w, v) \leq 2t \leq d-1.$$

This is a contradiction, since the distance between u and v is at least d .

Therefore,

$$B_t(u) \cap B_t(v) = \emptyset.$$

The total number of n -tuples is r^n , so

$$\sum_{u \in C} |B_t(u)| \leq r^n,$$

from which the required bound follows. ■

In contrast to packing spheres into a box in real space, in spaces over a finite alphabet it is possible that the whole space is filled. A block code which meets the sphere packing bound in [Theorem 3.9](#) is called a **perfect code**. For there to exist a perfect code we need parameters n , r and t so that

$$1 + \binom{n}{1}(r-1) + \dots + \binom{n}{t}(r-1)^t$$

divides r^n .

For example, it is possible that there is a perfect 2-error correcting binary code of length 90, a perfect 3-error correcting binary code of length 23, a perfect 2-error correcting ternary code of length 11. The former possibility is ruled out in [Exercise 3.3](#). However, as we shall see in [Chapter 5](#), the latter two perfect codes do occur, see [Example 5.9](#) and [Example 5.5](#).

The proof of the following lemma, [Lemma 3.10](#), counts in two ways the sum of the distances between any pair of codewords of an r -ary code. In the exercises related to [Lemma 3.10](#), [Exercise 3.4](#) treats a special case of equality in the bound and [Exercises 3.5](#) and [3.6](#) exploit the fact that if r does not divide $|C|$, then improvements can be made to the coordinate sum estimates.

Lemma 3.10 (Plotkin lemma) *An r -ary code C of length n and minimum distance d satisfies*

$$|C|(d + \frac{n}{r} - n) \leq d.$$

Proof

Let A be the alphabet, a set of r elements such that $C \subseteq A^n$. Fix $i \in \{1, \dots, n\}$ and let λ_a denote the number of times $a \in A$ occurs in the i -th coordinate of a codeword of C .

Clearly,

$$\sum_{a \in A} \lambda_a = |C|.$$

Since

$$\sum_{a \in A} (\lambda_a - |C|/r)^2 \geq 0,$$

we have

$$\sum_{a \in A} \lambda_a^2 - 2|C|^2/r + |C|^2/r \geq 0.$$

Let

$$S = \sum_{u, v \in C} d(u, v),$$

the sum of all distances between a pair (u, v) of codewords of C .

Let S_i be the contribution that the i -th coordinate makes to this sum.

Then,

$$S_i = \sum_{a \in A} \lambda_a (|C| - \lambda_a) \leq |C|^2 - |C|^2/r,$$

using the equality and the inequality from above.

Finally,

$$d|C|(|C| - 1) \leq S = \sum_{i=1}^n S_i \leq n(|C|^2 - |C|^2/r).$$

■

Example 3.11

Suppose that we wish to find a binary code C of length $2d - 2$ and minimum distance d . According to the bound in [Lemma 3.10](#), $|C| \leq d$. If we suppose that $|C| = d$, then we have equality in all the inequalities in the proof of [Lemma 3.10](#). In particular $\lambda_a = d/2$, which implies that d must be even. Moreover, the distance between any two codewords must be exactly d .

For $d = 4$, it is not difficult to find such a code, for example,

$$C = \{(0, 0, 0, 0, 0, 0), (1, 1, 1, 1, 0, 0), (1, 1, 0, 0, 1, 1), (0, 0, 1, 1, 1, 1)\}.$$

For $d = 6$ it is not so straightforward, see [Exercise 3.4](#).

■

[Lemma 3.10](#) only gives a bound on $|C|$ when $n < d + n/r$. If $n \geq d + n/r$ then, shortening the code by deleting coordinates so that for the shortened code $n' < d' + n'/r$, we can use [Lemma 3.10](#) repeatedly to obtain bounds for block codes with relatively smaller minimum distance. In [Theorem 3.12](#), we restrict to binary codes. For a general Plotkin bound on r -ary codes, see [Exercise 3.8](#). The bound in [Theorem 3.12](#) will be used to bound the rate of sequences of binary codes in [Section 3.3](#).

Theorem 3.12 (Plotkin bound)

If C is a binary code of length n and minimum distance $d \leq \frac{1}{2}n$, then

$$|C| \leq d2^{n-2d+2}.$$

Proof

Let $m = n - 2d + 1$. For each $x \in \{0, 1\}^m$, let C_x be the subset of C whose first m bits are the string x , where these m bits are then deleted. Then C_x is a block code of length $2d - 1$ and minimum distance $d + e$, for some $e \geq 0$. By [Lemma 3.10](#),

$$|C_x| \leq \frac{2(d+e)}{2e+1} \leq 2d.$$

Hence,

$$|C| = \sum_{x \in \{0,1\}^m} |C_x| \leq 2^m 2d = d2^{m-2d+2}.$$

■

3.3 Asymptotically Good Codes

If we want to send a large amount of data with short length codes, then we have to cut up a long string of n bits into strings of a fixed length n_0 . If the probability of decoding the string of length n_0 correctly is p , then the probability of decoding the string of length n is p^{n/n_0} , which tends to zero as n tends to infinity. Shannon's theorem, [Theorem 1.12](#), tells us that we should be able to send the string of n bits, through a channel with capacity Λ , which encodes almost Λn bits of information, and then decode correctly with a probability approaching 1. In the proof of Shannon's theorem, we used the fact the average number of errors which occur in the transmission of n bits through a binary symmetric channel is $(1 - \phi)n$. Therefore, our code of length n should be able to correct a number of errors which is linear in n . For this reason, in a sequence of codes of length n , we want that the minimum distance of the codes in the sequence also grows linearly with n .

A **sequence of asymptotically good codes** is a sequence of codes C_n of length n , where $n \rightarrow \infty$, in which d/n and $\log |C_n|/n$ are bounded away from zero. In other words, both the relative minimum distance and the rate are bounded away from zero.

For a sequence of asymptotically good binary codes C_n , define

$$R = \liminf(\log_2 |C_n|)/n.$$

For the remainder of this chapter, we will consider only binary codes and prove bounds on R . We start by applying the sphere packing bound to obtain an upper bound on R .

Theorem 3.13 (Sphere packing bound)

For a sequence of asymptotically good binary codes of relative minimum distance δ ,

$$R \leq 1 - h\left(\frac{1}{2}\delta\right).$$

Proof

Using Stirling's approximation $n! \sim (n/e)^n \sqrt{2\pi n}$,

$$\log \binom{n}{t} \sim n \log n - t \log t - (n-t) \log(n-t),$$

so

$$\frac{1}{n} \log \binom{n}{t} \sim \log n - \tau \log(\tau n) - (1 - \tau) \log((1 - \tau)n) = h(\tau),$$

where $\tau = t/n$.

Taking logarithms of both sides in the bound in [Theorem 3.9](#),

$$nR + \log \binom{n}{t} \leq n.$$

Therefore, for n large enough,

$$R + h(\tau) \leq 1,$$

which proves the bound, since $\tau = t/n = \lfloor (\delta n - 1)/2 \rfloor / n$. ■

Applying Plotkin's bound we can improve on this for larger values of δ , see [Figure 3.1](#).

Theorem 3.14 (Plotkin bound)

If $\delta \leq \frac{1}{2}$, then

$$R \leq 1 - 2\delta.$$

Proof

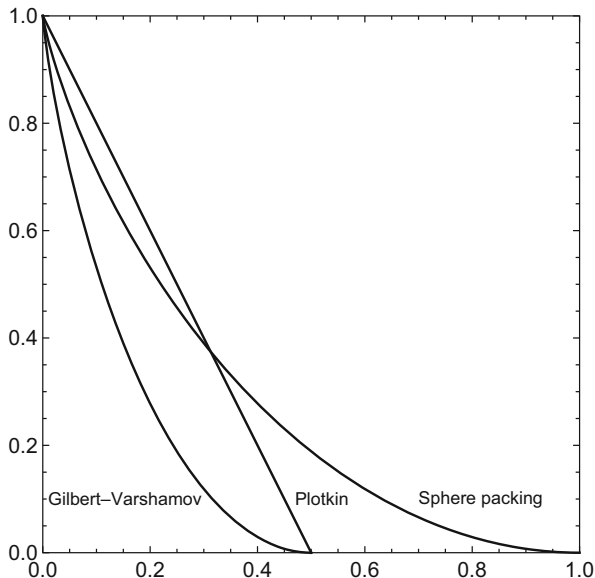
By [Theorem 3.12](#),

$$2^{Rn} \leq \delta n 2^{n-2\delta n+2}.$$

Taking logarithms, dividing by n and letting n tend to infinity, the bound follows. ■

Our aim now is to improve on Plotkin's bound. The main idea behind the proof of [Theorem 3.16](#) is to bound the number of codewords at a fixed distance w from a fixed n -tuple v . By changing 0 to 1 and 1 to 0 in the coordinates where v has a 1 (applying the change to all codewords) we obtain a code with the same parameters (length n and minimum distance d) in which v is now the all-zero n -tuple. Then the number of codewords at distance w to v is the number of codewords of weight w , where the **weight** of $u \in \{0, 1\}^n$, denoted $\text{wt}(u)$, is the number of non-zero coordinates that u has. Therefore, [Lemma 3.15](#) is really bounding the number of codewords at a fixed distance w to a fixed n -tuple v . In [Exercise 3.12](#), we will bound the number of codewords at distance at most w to a fixed n -tuple v . This is an important bound because it tells us that although we cannot in general correct more than $\frac{1}{2}d$ errors, even if more errors occur, there are relatively few codewords

Fig. 3.1 Plotting of the bounds relative minimum distance (x) against rate (y).



which could have been sent, providing that not more than $\frac{1}{2}n(1 - \sqrt{1 - 2(d/n)})$ errors occur. This opens up the possibility of a list decoding algorithm which creates a short list of possible codewords which could have been sent, even in the case that the received n -tuple contains a large amount of errors. It is also possible that the list contains just one codeword. Although $e \geq \frac{1}{2}d$ errors occur in the transmission, it may be the case that there is only one codeword at distance e from the received n -tuple. It is even more useful if we use two codes simultaneously. Take two codes with high rates R_1 and R_2 , which code the same sequence of bits and suppose both codes are **systematic**, which means that they conserve the original message and add some check bits. [Example 3.4](#) is an example of a systematic code and, as we shall see, so is a linear code with a generator matrix in standard form. Consider the code we obtain by sending the original message and the two sets of check bits. This code has rate

$$\frac{R_1 R_2}{R_1 + R_2 - R_1 R_2}.$$

If we have a list decoding algorithm for both codes, then the sent codeword will be in the intersection of the two lists and with a high probability will be the only candidate in the intersection.

Let $A(n, d, w)$ denote the maximum size of a binary code of length n and minimum distance d in which each codeword has exactly w ones.

Lemma 3.15 *The following inequality holds*

$$A(n, d, w) \leq \frac{nd}{2w^2 - 2wn + dn}.$$

Proof

Let C be a binary code of length n and minimum distance d of size $A(n, d, w)$ in which all codewords have weight w . For any two codewords $u, v \in C$, $d(u, v) \geq d$, so there are at least d coordinates in which they differ. One of the two codewords has ones in at least $\frac{1}{2}d$ of these coordinates. Hence, the scalar product

$$u \cdot v \leq w - \frac{1}{2}d.$$

Summing over all pairs of codewords we get

$$\sum_{u, v \in C} u \cdot v \leq (w - \frac{1}{2}d)|C|(|C| - 1).$$

Let λ_i denote the number of times one appears in the i -th coordinate of a codeword of C .

Counting in two ways the number of triples (u_i, v_i, i) where u and v are codewords such that $u_i = v_i = 1$ and $i \in \{1, \dots, n\}$,

$$\sum_{i=1}^n \lambda_i(\lambda_i - 1) = \sum_{u, v \in C} u \cdot v.$$

Since every codeword of C has exactly w ones,

$$\sum_{i=1}^n \lambda_i = w|C|.$$

The sum

$$\sum_{i=1}^n (\lambda_i - \frac{w}{n}|C|)^2 \geq 0,$$

which implies that

$$\sum_{i=1}^n \lambda_i^2 \geq \frac{w^2}{n}|C|^2.$$

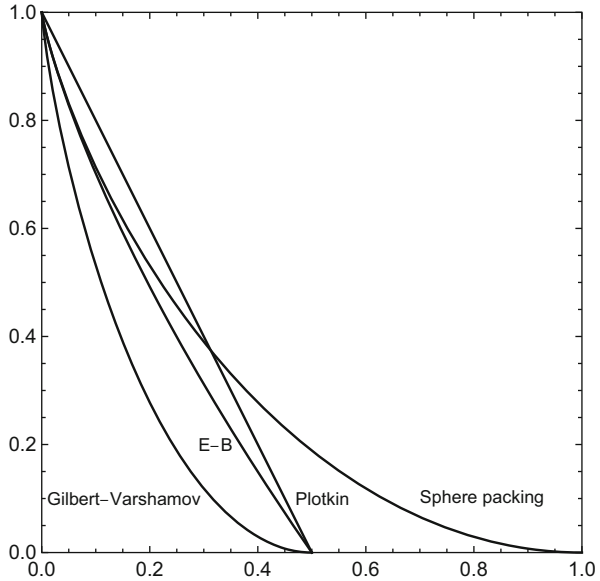
Hence,

$$\frac{w^2}{n}|C|^2 - w|C| \leq (w - \frac{1}{2}d)|C|(|C| - 1),$$

from which the bound follows. ■

The bound in the following theorem improves on the Plotkin bound and the sphere packing bound, see [Figure 3.2](#). Observe that if we had a better bound on $A(n, d, w)$, then we could improve the bound on R . Recall that δ is the relative minimum distance.

Fig. 3.2 Plotting the bounds of relative minimum distance (x) against rate (y).



Theorem 3.16 (Elias–Bassalygo bound)

If $\delta < \frac{1}{2}$, then

$$R \leq 1 - h\left(\frac{1}{2}(1 - \sqrt{1 - 2\delta})\right).$$

Proof

Let C be a binary code of length n and minimum distance $d = \lfloor \delta n \rfloor$.

For a fixed $w \in \{1, \dots, n\}$, codeword $u \in C$ and $v \in \{0, 1\}^n$, let

$$b(u, v) = \begin{cases} 1 & d(u, v) = w, \\ 0 & \text{otherwise.} \end{cases}$$

Then,

$$\sum_{u \in C} \sum_{v \in \{0,1\}^n} b(u, v) = \sum_{u \in C} \binom{n}{w} = \binom{n}{w} |C|.$$

For a fixed v , let C' be the set of codewords of C at distance w to v . If we interchange 0 and 1 in the coordinates where v has a 1, we obtain a code from C' of constant weight w . The number of codewords u for which $b(u, v) = 1$ is $|C'|$, which is at most $A(d, n, w)$.

Switching the order of the summations in the above equality implies

$$\sum_{v \in \{0,1\}^n} \sum_{u \in C} b(u, v) \leq 2^n A(n, d, w).$$

Hence,

$$\binom{n}{w} |C| \leq 2^n A(n, d, w).$$

By [Lemma 3.15](#),

$$(2w^2 - 2wn + dn) \binom{n}{w} |C| \leq nd2^n.$$

Now, choose $w \in \mathbb{N}$ so that

$$\frac{2w}{n} \approx 1 - \sqrt{1 - 2\frac{d}{n} + \frac{2}{n}}.$$

Then,

$$2w^2 - 2wn + dn \approx 2(w - \frac{1}{2}n)^2 - \frac{1}{2}n^2 + dn \approx n.$$

As in the proof of [Theorem 3.13](#), using Stirling's approximation,

$$\frac{1}{n} \log_2 \binom{n}{w} > h(w/n),$$

for n large enough. Taking logarithms of

$$\binom{n}{w} |C| \leq d2^n,$$

gives

$$\log_2 |C| + nh(w/n) \leq n + \log_2 d.$$

Dividing by n , and letting n tend to infinity, we get the bound. ■

The following bound is an improvement to Elias–Bassalygo bound for $\delta > 0.14$. We include it here without a proof. There are other improvements to Elias–Bassalygo bound known but they are quite complicated, even to state.

Theorem 3.17 (McEliece–Rodemich–Rumsey–Welch bound)

$$R \leq h\left(\frac{1}{2} - \sqrt{\delta(1-\delta)}\right).$$

In [Figure 3.3](#), the Gilbert–Varshamov lower bound is compared to the best upper bounds we have seen, the Elias–Bassalygo bound and the McEliece–Rodemich–Rumsey–Welch bound. There has been little progress in closing the gap between the lower and upper bounds. The following conjecture is one of the oldest and yet still unresolved conjectures in coding theory.

Conjecture 3.18 Given δ , the Gilbert–Varshamov bound gives the maximum rate for binary codes with relative minimum distance δ as $n \rightarrow \infty$.

The following conjecture has been open since 1973.

Conjecture 3.19 Apart from trivial examples, there are no perfect constant weight binary codes, i.e. there are no constant weight codes achieving the bound in [Exercise 3.9](#).

Although [Theorem 3.7](#) tells us that asymptotically good codes exist, there is a fundamental issue which needs to be addressed. We have no feasible algorithm which allows us to decode such codes. The best we can do, for such a randomly chosen code, is to simply go

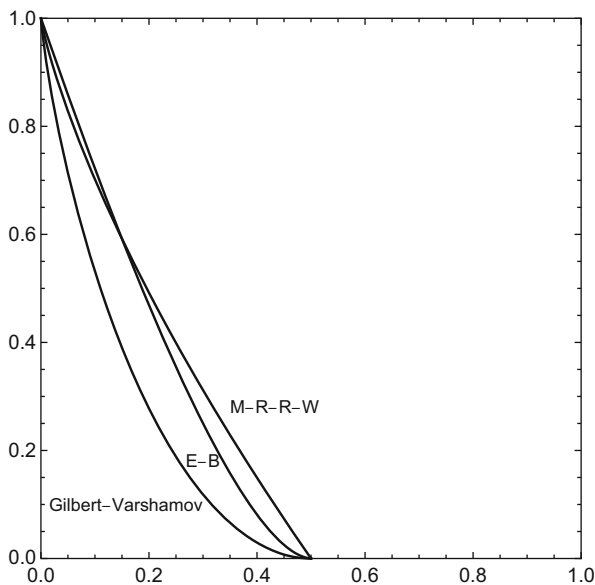


Fig. 3.3 Plotting of the bounds relative minimum distance (x) against rate (y).

through the codewords, one-by-one, until we find a codeword which is at a distance at most t to the received vector. This is laboriously slow and almost never efficient. In fact, we do not even have a fast algorithm to assign the codewords. In the next few chapters we shall start to address these issues.

3.4 Comments

There are many texts which cover block codes, the books by Bierbrauer [10], Hill [37], Roman [61], Berlekamp [8], MacWilliams and Sloane [50], Ling and Xing [48] and Van Lint [74] being some classical examples. They are all interesting reads and give more insight into coding theory. In this book we cover much of the material common to these books and will progress to topics which have been developed since their publication.

The Gilbert–Varshamov bound is from Gilbert [25], Varshamov proving a similar bound for linear codes in [75], see Exercise 4.3. There are no known explicit constructions of asymptotically good codes meeting the Gilbert–Varshamov bound, unless the relative minimum distance is close to $\frac{1}{2}$, see Ta-Shma [69]. There are many improvements to the Gilbert–Varshamov bound known but these do not outdo the bound asymptotically.

The Elias–Bassalygo bound is from [7]. The Plotkin bound is from [58] and the McEliece, Rodemich, Rumsey and Welch bound highlighted in the text is from [51]. There are bounds of Delsarte which better these bounds for some values of δ , see [21].

Conjecture 3.19 is from Delsarte’s thesis, see [21]. The main result of Roos [62] restricts the range of parameters for which equality can occur.

3.5 Exercises

3.1 Prove that if we use the repetition code and nearest neighbour decoding for the binary symmetric channel in which the probability that a bit changes is $1 - \phi < \frac{1}{2}$, then $P_{\text{COR}} \rightarrow 1$ as $n \rightarrow \infty$.

3.2 Prove that

$$C = \{(x_1, x_2, x_3, x_4, x_1 + x_2 + x_3, x_1 + x_2 + x_4, x_1 + x_3 + x_4) \mid x_1, x_2, x_3, x_4 \in \mathbb{F}_2\}$$

is a perfect 1-error correcting binary code of length 7.

3.3 Prove that there is no perfect 2-error correcting binary code of length 90.

3.4 Construct a binary code of length 10 and minimum distance 6 of size 6 and so prove that $A_2(10, 6) = 6$. Use the solution to prove that the bound in Lemma 3.15 is attainable for $A_2(10, 6, 6)$.

3.5

- i. Prove that there is no ternary code of length 6, minimum distance 5 of size 5.
- ii. Suppose that C is a ternary code of length 6, minimum distance 5 of size 4. Prove that for every symbol $x \in \{0, 1, 2\}$ and every coordinate i , there is a codeword in C with an x in the i -th coordinate.
- iii. Construct a ternary code of length 6, minimum distance 5 of size 4 and conclude that $A_3(6, 5) = 4$.

3.6 Prove that $A_2(8, 5) = 4$.

3.7 Let C be the code obtained from two systematic codes of rates R_1 and R_2 where the two sets of check bits are appended to the message bits. Prove that the rate of C is

$$\frac{R_1 R_2}{R_1 + R_2 - R_1 R_2}.$$

3.8 Prove that if $n \geq dr/(r-1)$, then an r -ary code C satisfies

$$|C| \leq \frac{dr^{m+1}}{dr - (n-m)(r-1)},$$

where m is such that $m > n - rd/(r-1)$.

3.9 Prove the sphere packing bound for binary constant weight codes,

$$\left(\sum_{i=0}^e \binom{n}{i} \binom{n-w}{i} \right) A(n, 4e+2, w) \leq \binom{n}{w}.$$

3.10 Prove that if d is odd, then

$$A(n, d, w) \leq \frac{dn+n}{2w^2 - 2wn + dn + n}.$$

3.11 Prove that $A(8, 5, 5) = 2$.

3.12

- i. Prove that if $v_1, \dots, v_r \in \mathbb{R}^n$ have the property that $v_i \cdot v_j \leq 0$, then $r \leq 2n$.
- ii. For $u = (u_1, \dots, u_n) \in \{0, 1\}^n$, let $\sigma(u)$ be the vector whose j -th coordinate is $(-1)^{u_j}$. Prove that

$$\sigma(u) \cdot (1, 1, \dots, 1) = n - 2\text{wt}(u).$$

3.5 · Exercises

iii. For $u, v \in \{0, 1\}^n$, prove that

$$\sigma(u) \cdot \sigma(v) = n - 2d(u, v).$$

iv. Let C be a binary code of length n and minimum distance d . Prove that for $u, v \in C$

$$(\sigma(u) - \lambda(1, 1, \dots, 1)) \cdot (\sigma(v) - \lambda(1, 1, \dots, 1)) \leq \lambda^2 n + 2\lambda(2w - n) + n - 2d.$$

v. By choosing an appropriate λ in iv., prove that if $w \leq \frac{1}{2}n(1 - \sqrt{1 - 2(d/n)})$, then

$$\sum_{j=0}^w A(n, d, j) \leq 2n.$$



Linear Codes

There is a lack of structure in the block codes we have considered in the first few chapters. Either we chose the code entirely at random, as in the proof of [Theorem 1.12](#), or we built the code using the greedy algorithm, as in the proof of the Gilbert–Varshamov bound, [Theorem 3.7](#). In this chapter, we introduce some algebraic structure to the block codes by restricting our attention to linear codes, codes whose codewords are the vectors of a subspace of a vector space over a finite field. Linear codes have the immediate advantage of being fast to encode. We shall also consider a decoding algorithm for this broad class of block codes. We shall prove the Griesmer bound, a bound which applies only to linear codes and show how certain linear codes can be used to construct combinatorial designs.

4.1 Preliminaries

If $A = \mathbb{F}_q$ and C is a subspace of \mathbb{F}_q^n , then we say that C is a **linear code over \mathbb{F}_q** or simply a **linear code**. If the subspace has dimension k , then we say that C is a **k -dimensional linear code over \mathbb{F}_q** . Observe that $|C| = q^k$.

As in the case of an n -tuple, we define the **weight** $\text{wt}(v)$ of a vector $v \in \mathbb{F}_q^n$ as the number of non-zero coordinates that v has. Recall that the elements of a code are called codewords.

Lemma 4.1 *The minimum distance d of a linear code C is equal to the minimum weight w of a non-zero codeword of C .*

Proof

Suppose $u \in C$ is a codeword of minimum non-zero weight w . Since C is a subspace, the zero vector 0 is in C . Clearly $d(u, 0) = w$, so $w \geq d$.

Suppose u and v are two codewords at minimum distance from each other, so $d(u, v) = d$. Since C is linear, $u - v \in C$, and $d(u - v, 0) = d$. Hence, there is a codeword in C with weight d , which implies that $d \geq w$. \square

We can describe a linear code C by means of a basis. A matrix G whose rows are a basis for C is called a **generator matrix** for C . Thus,

$$C = \{vG \mid v \in \mathbb{F}_q^k\}.$$

We will often use the short-hand notation $[n, k, d]_q$ code to mean that the code is a k -dimensional linear code of length n and minimum distance d over \mathbb{F}_q . For a not necessarily linear code, we use the notation $(n, K, d)_r$ code to mean a code of length n , minimum distance d of size K over an alphabet of size r .

Example 4.2

The minimum weight of the non-zero codewords of the 4-dimensional linear code of length 7 over \mathbb{F}_2 generated by the matrix

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

is 3, so it follows from [Lemma 4.1](#) that the code is a $[7, 4, 3]_2$ code. ■

Example 4.3

Consider the $[9, 3, d]_3$ code generated by the matrix

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

Each row of G has weight 6 and it is immediate to verify that a linear combination of two of the rows also has weight 6. Any linear combination of the first two rows has at most 3 coordinates in common with the third row, so we can conclude that the minimum weight of a non-zero codeword is 6. By [Lemma 4.1](#), G is the generator matrix of a $[9, 3, 6]_3$ code. ■

We can also define a linear code as the solution of a system of linear equations. A **check matrix** for a linear code C is an $m \times n$ matrix H with entries from \mathbb{F}_q , with the property that

$$C = \{u \in \mathbb{F}_q^n \mid uH^t = 0\},$$

where H^t denotes the transpose of the matrix H .

Lemma 4.4 *Let C be a linear code with check matrix H . If every set of $d - 1$ columns of H are linearly independent, and some set of d columns are linearly dependent, then C has minimum distance d .*

Proof

Let u be a codeword of C and let D be the set of non-zero coordinates of u , so $|D| = \text{wt}(u)$. Let h_i be the i -th column of H . Since H is a check matrix for C ,

$$\sum_{i \in D} u_i h_i = 0.$$

Thus, there is a linear combination of $|D|$ columns of H which are linearly dependent. Applying [Lemma 4.1](#) concludes the proof. \square

Example 4.5

Let C be the linear code over \mathbb{F}_q defined by the $m \times n$ check matrix H , whose columns are vectors which span distinct one-dimensional subspaces of \mathbb{F}_q^m . In other words, the columns of H are vector representatives of distinct points of $\text{PG}(m-1, q)$. Since any two columns of H are linearly independent, [Lemma 4.4](#) implies that C has minimum distance at least 3. By [Exercise 2.11](#), the number of points of $\text{PG}(m-1, q)$ is $(q^m - 1)/(q - 1)$, so

$$n \leq (q^m - 1)/(q - 1).$$

If we take

$$n = (q^m - 1)/(q - 1),$$

then C is a code of size q^k with parameters, $d = 3$ and

$$k = (q^m - 1)/(q - 1) - m.$$

This code C attains the bound in [Theorem 3.9](#), since

$$|C|(1 + n(q - 1)) = q^k(1 + q^m - 1) = q^n.$$

Thus, C is a perfect code. \blacksquare

[Example 4.5](#) is called the **Hamming code**. [Example 4.2](#) is the Hamming code with $q = 2$ and $m = 3$. A check matrix for this code is

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

One can readily check that $GH^t = 0$, where G is as in [Example 4.2](#), by verifying that the scalar product of any row of G with any row of H is zero.

Lemma 4.6 Let G be a generator matrix for a k -dimensional linear code C . An $m \times n$ matrix H is a check matrix for C if and only if $GH^t = 0$ and the rank of H is $n - k$.

Proof

Suppose that H is an $m \times n$ check matrix for C . All the rows of G are codewords of C , so if u is a row of G , then $uH^t = 0$, which implies $GH^t = 0$. The dimension of the code C is $n - \text{rank}(H)$, which implies that the rank of H is $n - k$.

Suppose that $GH^t = 0$ and that the rank of H is $n - k$. A codeword u is a linear combination of the rows of G , so $uH^t = 0$. Hence, the left kernel of H^t contains C . Since the rank of H is $n - k$, the left kernel of H^t has dimension k , so the left kernel of H^t is C . \square

Let I_r denote the $r \times r$ identity matrix.

A generator matrix which has the form

$$(I_k \mid A),$$

for some $k \times (n - k)$ matrix A , is said to be in **standard form**. The uncoded string v is encoded by vG , whose first k coordinates are precisely the coordinates of v . There are obvious advantages in using a generator matrix in this standard form. Once errors have been corrected, the uncoded string can be recovered from the codeword by simply deleting the last $n - k$ coordinates. Moreover, the following lemma implies that there is a check matrix with a similar simple form.

Lemma 4.7 Let C be the linear code generated by

$$G = (I_k \mid A),$$

for some $k \times (n - k)$ matrix A . Then the matrix

$$H = (-A^t \mid I_{n-k})$$

is a check matrix for C .

Proof

We have to check that the inner product of the i -th row of $G = (g_{ij})$ with the ℓ -th row of $H = (h_{\ell j})$ is zero. The entries $g_{ij} = 0$ for $j \leq k$ unless $i = j$, in which case $g_{ii} = 1$. The entries $h_{\ell j} = 0$ for $j \geq k + 1$ unless $\ell = j - k$, in which case $h_{\ell, \ell+k} = 1$. Hence,

$$\sum_{j=1}^n g_{ij}h_{\ell j} = \sum_{j=1}^k g_{ij}h_{\ell j} + \sum_{j=k+1}^n g_{ij}h_{\ell j} = h_{\ell i} + g_{i, \ell+k} = -a_{i\ell} + a_{i\ell} = 0.$$

\square

4.2 Syndrome Decoding

Given a generator matrix G for a linear code C , encoding is fairly simple since we assign the codeword vG to each vector v of \mathbb{F}_q^k . Moreover, if the generator matrix is in standard form, as described in the previous section, then we can encode by appending the $n - k$ coordinates of vA to v . Decoding is a far trickier affair. To use nearest neighbour decoding we have to find the codeword of length n which is nearest to the received n -tuple. For a code with no obvious structure, this can only be done by calculating the distance between the received n -tuple and each codeword, something which is laborious and infeasible for large codes. In this section, we consider a decoding algorithm for linear codes which exploits the linearity property.

Let C be a linear code with check matrix H . The **syndrome** of a vector $v \in \mathbb{F}_q^n$ is

$$s(v) = vH^t.$$

Note that $s(v) = 0$ if and only if $v \in C$, since

$$C = \{v \in \mathbb{F}_q^n \mid vH^t = 0\}.$$

To use **syndrome decoding** we compute a look-up table with entries $s(e)$ for all vectors e of weight at most $t = \lfloor (d - 1)/2 \rfloor$. To decode a vector v we compute $s(v)$, use the look-up table to find e such that $s(v) = s(e)$, and decode v as $v - e$. Note that $v - e \in C$ and the distance between v and $v - e$ is at most t .

Example 4.8

The matrix

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

is the generator matrix of a $[8, 4, 4]_3$ code.

Suppose that a codeword u has been sent and we have received the vector

$$v = (1, 0, 1, 0, 0, 1, 0, 2).$$

By [Lemma 4.7](#), the matrix

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

is a check matrix for C .

Note that $-1 = 2$, since we are doing arithmetic with elements of \mathbb{F}_3 . To decode using syndrome decoding, we calculate the syndrome of v ,

$$s(v) = vH^t = (2, 2, 2, 0).$$

Then we look for the low weight vector e , in this example a vector of weight one, such that $s(v) = s(e)$. If only one error has occurred in the transmission, the syndrome $s(v)$ must be equal to $s(e)$, for some vector e of \mathbb{F}_q^8 of weight one. Indeed,

$$s(v) = s((0, 0, 0, 1, 0, 0, 0, 0)).$$

Therefore, we correct v to the codeword

$$v - e = (1, 0, 1, 2, 0, 1, 0, 2),$$

which is $(1, 0, 1, 2)G$. ■

In general, using a look-up table would involve searching through

$$\sum_{j=1}^t \binom{n}{j} (q-1)^j$$

entries, an entry for each non-zero vector of \mathbb{F}_q^n of weight at most t . For n large, this implies that we would have to search through a table with an exponential number of entries, since

$$\binom{n}{\frac{1}{2}\delta n} \sim 2^{h(\frac{1}{2}\delta)n}.$$

This does not imply that there might not be a better method to find the vector e with the property that $s(e) = s(v)$, especially if the linear code has some additional properties we can exploit. However, we will now prove that decoding a linear code using syndrome decoding is an NP problem. Under the assumption that $P \neq NP$, this implies that there is no polynomial time algorithm that will allow us to decode using syndrome decoding.

Problems in NP are, by definition, decision problems. So what we mean by saying that decoding a linear code using syndrome decoding is an NP problem, is that deciding if we can decode a linear code using syndrome decoding is an NP problem. A decision problem is in P if there exists a polynomial time algorithm which gives a yes/no answer to the problem. A decision problem is in NP, if there exists a polynomial time algorithm which verifies that a “yes” solution to the problem, really is a solution. For example, the Hamiltonian path problem asks if there is a path in a graph which visits all the vertices without repeating any vertex. This is an NP problem since a “yes” solution to the problem is a Hamiltonian path. This solution can be checked in polynomial time by checking that each edge in the path is an edge of the graph.

It is not known if NP is a larger class of problems than P or not. A decision problem D is said to be NP-**complete** if there is a polynomial time algorithm which reduces every problem in NP to D. This implies that if we had a polynomial time algorithm to solve D, then we would have a polynomial time algorithm to solve all problems in NP.

Let T be a subset of $\{1, \dots, n\}^3$.

A **perfect matching** M is a subset of T of size n ,

$$M = \{(a_{j1}, a_{j2}, a_{j3}) \mid j = 1, \dots, n\} \subseteq T,$$

where for all $i \in \{1, 2, 3\}$,

$$\{a_{ji} \mid j = 1, \dots, n\} = \{1, \dots, n\}.$$

Deciding whether T has a perfect matching or not is the **three-dimensional matching problem**. This decision problem is NP-complete.

For example, let T be the set of triples

$$\{(1, 1, 1), (1, 2, 3), (1, 4, 2), (2, 1, 4), (2, 3, 3), (3, 2, 1), (3, 3, 4), \\ (4, 3, 2), (4, 3, 3), (4, 4, 4)\}.$$

The three-dimensional matching problem asks if it is possible to find a subset M of T such that each element of $\{1, 2, 3, 4\}$ appears in each coordinate of an element of M exactly once. In this example the answer is affirmative,

$$M = \{(1, 4, 2), (2, 1, 4), (3, 2, 1), (4, 3, 3)\}.$$

Theorem 4.9

Decoding a linear code using syndrome decoding is NP-complete.

Proof

To decode a linear code using syndrome decoding, we have to find a vector e of weight at most t , such that $eH^t = s$, where $s = s(v)$ and v is the received vector.

We make this a decision problem by asking if there is a vector e of weight at most t such that $eH^t = s$. We will show that this decision problem is NP-complete by proving that if we had a polynomial time algorithm to solve this decision problem, then we would have a polynomial time algorithm to solve the three-dimensional matching problem.

Let $R_i = \{1, \dots, n\}$ for $i = 1, 2, 3$. Let T be a subset of $R_1 \times R_2 \times R_3$. Consider the matrix A whose rows are indexed by the triples in T , whose columns are indexed by $R_1 \cup R_2 \cup R_3$, where the $((a_1, a_2, a_3), r_i)$ entry is 1 if $a_i = r_i$ and zero otherwise. Thus, each row has three ones and $3n - 3$ zeros. A perfect matching is given by a vector v of $\{0, 1\}^{|T|}$, necessarily of weight n , such that vA is equal to the all-one vector j . Therefore, if we have a

polynomial time algorithm which can decide if there is a vector e of weight at most t , such that $eH^t = s$, then we can use this to solve the three-dimensional perfect matching decision problem by asking if there is a vector v of weight n such that $vA = j$. \square

4.3 Dual Code and the MacWilliams Identities

Let C be a k -dimensional linear code over \mathbb{F}_q .

The **dual code** of a linear code C is

$$C^\perp = \{v \in \mathbb{F}_q^n \mid u \cdot v = u_1v_1 + \cdots + u_nv_n = 0, \text{ for all } u \in C\}.$$

In other words C^\perp is the orthogonal subspace to C , with respect to the standard inner product. The subspace C^\perp is the set of solutions of a homogeneous system of linear equations of rank k in n unknowns. Hence, the dual code C^\perp is a $(n - k)$ -dimensional linear code and length n over \mathbb{F}_q .

The following lemma is immediate.

Lemma 4.10 *If H is a $(n - k) \times n$ check matrix for a k -dimensional linear code C , then H is a generator matrix for C^\perp . Likewise, if G is a generator matrix for C , then G is a check matrix for C^\perp .*

If $C = C^\perp$, then we say that C is **self-dual**.

Example 4.11

The extended code of the binary four-dimensional code in [Example 4.2](#) is a self-dual code. It has a generator (and check) matrix

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

■

Let A_i denote the number of codewords of weight i of a linear code C of length n . The **weight enumerator** of C is a polynomial defined as

$$A(X) = \sum_{i=0}^n A_i X^i.$$

Let $A^\perp(X)$ denote the weight enumerator of the dual code C^\perp .

There is an important relationship between $A(X)$ and $A^\perp(X)$, which implies that one is determined by the other. To be able to prove this relationship, which we shall do in [Theorem 4.13](#), we introduce the trace map and characters.

Let p be the prime such that $q = p^h$. Then the **trace map** from \mathbb{F}_q to \mathbb{F}_p is defined as

$$\text{Tr}(x) = x + x^p + \cdots + x^{q/p}.$$

By [Lemma 2.6](#), it is additive, i.e.

$$\text{Tr}(x + y) = \text{Tr}(x) + \text{Tr}(y),$$

and by [Lemma 2.4](#) and [Lemma 2.6](#),

$$\text{Tr}(x)^p = \text{Tr}(x),$$

so, again by [Lemma 2.4](#), $\text{Tr}(x) \in \mathbb{F}_p$.

Observe that if $\text{Tr}(\lambda x) = 0$ for all $\lambda \in \mathbb{F}_q$, then $x = 0$, since as a polynomial (in λ) it has degree q/p . For the same reason, every element of \mathbb{F}_p has exactly q/p pre-images of the trace map from \mathbb{F}_q to \mathbb{F}_p .

For $u \in \mathbb{F}_q^n$, we define a **character** as a map from \mathbb{F}_q^n to \mathbb{C} by

$$\chi_u(x) = e^{\frac{2\pi i}{p} \text{Tr}(x \cdot u)}.$$

Note that this definition makes sense since \mathbb{F}_p is $\mathbb{Z}/(p\mathbb{Z})$.

Lemma 4.12 *Let C be a linear code over \mathbb{F}_q . Then*

$$\sum_{u \in C} \chi_u(x) = \begin{cases} 0 & \text{if } x \notin C^\perp \\ |C| & \text{if } x \in C^\perp \end{cases}.$$

Proof

If $x \in C^\perp$, then $x \cdot u = 0$ for all $u \in C$ which implies $\chi_u(x) = 1$ for all $u \in C$ and we are done.

Suppose $x \notin C^\perp$. If $\chi_v(x) = 1$ for all $v \in C$, then $\text{Tr}(v \cdot x) = 0$ for all $v \in C$, so $\text{Tr}(\lambda \cdot x) = 0$ for all $\lambda \in \mathbb{F}_q$ and $v \in C$. This, we observed before, implies $v \cdot x = 0$ for all $v \in C$, so $x \in C^\perp$, a contradiction. Thus, there is a $v \in C$ such that $\chi_v(x) \neq 1$. Then,

$$\chi_v(x) \sum_{u \in C} \chi_u(x) = \sum_{u \in C} \chi_{u+v}(x) = \sum_{u \in C} \chi_u(x).$$

which implies

$$\sum_{u \in C} \chi_u(x) = 0.$$

□

The following theorem relates the weight enumerator of a linear code to the weight enumerator of its dual code. It is known as the **MacWilliams identities**.

Theorem 4.13 (MacWilliams)

For a k -dimensional linear code C over \mathbb{F}_q of length n we have

$$q^k A^\perp(X) = (1 + (q - 1)X)^n A\left(\frac{1 - X}{1 + (q - 1)X}\right).$$

Proof

Let $u = (u_1, \dots, u_n) \in \mathbb{F}_q^n$.

If $u_i \neq 0$, then

$$\sum_{w_i \in \mathbb{F}_q} \chi_{w_i e_i}(u) = 0,$$

since we sum each p -th root of unity q/p times, and the sum of the p -th roots of unity is zero.

Therefore,

$$\sum_{w_i \in \mathbb{F}_q \setminus \{0\}} \chi_{w_i e_i}(u) = \begin{cases} q - 1 & \text{if } u_i = 0 \\ -1 & \text{if } u_i \neq 0 \end{cases}$$

and so

$$\prod_{i=1}^n \left(1 + \sum_{w_i \in \mathbb{F}_q \setminus \{0\}} \chi_{w_i e_i}(u) X\right) = (1 + (q - 1)X)^{n - \text{wt}(u)} (1 - X)^{\text{wt}(u)}.$$

Multiplying out the brackets,

$$\prod_{i=1}^n \left(1 + \sum_{w_i \in \mathbb{F}_q \setminus \{0\}} \chi_{w_i e_i}(u) X\right) = \sum_{w \in \mathbb{F}_q^n} X^{\text{wt}(w)} \prod_{i=1}^n \chi_{w_i e_i}(u) = \sum_{w \in \mathbb{F}_q^n} X^{\text{wt}(w)} \chi_u(w).$$

Combining the above two equations,

$$\sum_{w \in \mathbb{F}_q^n} X^{\text{wt}(w)} \chi_u(w) = (1 + (q - 1)X)^{n - \text{wt}(u)} (1 - X)^{\text{wt}(u)}.$$

Summing over $u \in C$, we have

$$\sum_{u \in C} \sum_{w \in \mathbb{F}_q^n} X^{\text{wt}(w)} \chi_u(w) = (1 + (q - 1)X)^n A\left(\frac{1 - X}{1 + (q - 1)X}\right),$$

since

$$A(X) = \sum_{u \in C} X^{\text{wt}(u)}.$$

Switching the order of the summations, and applying [Lemma 4.12](#),

$$\sum_{w \in \mathbb{F}_q^n} X^{\text{wt}(w)} \sum_{u \in C} \chi_u(w) = \sum_{w \in C^\perp} X^{\text{wt}(w)} |C| = |C| A^\perp(X).$$

□

Observe that [Theorem 4.13](#) implies that if we know the weights of the codewords of C , then we know the weights of the codewords of C^\perp and in particular the minimum weight of a non-zero codeword and therefore, by [Lemma 4.1](#), the minimum distance of C^\perp .

If C is a self-dual code, we can get information about the weights of the codewords of C from [Theorem 4.13](#).

Example 4.14

Let C be a self-dual 4-dimensional binary linear code of length 8, for instance, as in [Example 4.11](#). Then, equating the coefficient of X^j , for $j = 0, \dots, 8$, in

$$A(X) = 2^{-4}(1+X)^8 A((1-X)/(1+X)),$$

where

$$A(X) = 1 + \sum_{i=1}^8 a_i X^i,$$

will give a system of nine linear equations and eight unknowns.

This system has the solution

$$A(X) = 1 + 14X^4 + X^8 + \lambda(X^2 - 2X^4 + X^6),$$

for some $\lambda \in \{0, \dots, 7\}$. Thus, C must contain the all-one vector and if the minimum distance of C is 4, then

$$A(X) = 1 + 14X^4 + X^8.$$

■

We will see an important application of the MacWilliams identities in [Section 4.6](#) where we will exploit these equations to prove that, under certain hypotheses, we can construct combinatorial designs from a linear code.

4.4 Linear Codes and Sets of Points in Projective Spaces

A linear code C is the row space of a generator matrix G . The multi-set S of columns of G also contains information about the code and its parameters. The length of C is $|S|$, the dimension of C is the length of the vectors in S and, as we shall prove in [Lemma 4.15](#), the weights of the codewords in C can be deduced from the intersection of S with the hyperplanes of \mathbb{F}_q^k . Observe that S is a multi-set since columns can be repeated.

Lemma 4.15 *The multi-set S of columns of a generator matrix G of a $[n, k, d]_q$ code C is a multi-set of n vectors of \mathbb{F}_q^k in which every hyperplane of \mathbb{F}_q^k contains at most $n - d$ vectors of S , and some hyperplane of \mathbb{F}_q^k contains exactly $n - d$ vectors of S .*

Proof

There is a bijection between the vectors of \mathbb{F}_q^k and the codewords, given by

$$v \mapsto vG.$$

For each non-zero vector v of \mathbb{F}_q^k , the subspace consisting of the vectors $(x_1, \dots, x_k) \in \mathbb{F}_q^k$, such that

$$v_1x_1 + \dots + v_kx_k = 0,$$

is a hyperplane of \mathbb{F}_q^k , which we denote by π_v . The non-zero multiples of v define the same hyperplane, so $\pi_v = \pi_{\lambda v}$, for all non-zero $\lambda \in \mathbb{F}_q$.

We can label the coordinates of vG by the elements of S . The s -coordinate of the codeword vG is the value of the scalar product $v \cdot s$. The scalar product $v \cdot s = 0$ if and only if $s \in \pi_v$. Therefore, the codeword vG has weight w if and only if the hyperplane π_v contains $n - w$ vectors of S . The lemma follows since, by [Lemma 4.1](#), the minimum weight of a non-zero vector of C is equal to the minimum distance. \square

[Lemma 4.15](#) is still valid if we replace a vector s of S by a non-zero scalar multiple of s . Thus, we could equivalently state the lemma for a multi-set of points in $\text{PG}(k - 1, q)$, assuming that the vectors in S are non-zero vectors. In the projective space, the hyperplane π_v is a hyperplane of $\text{PG}(k - 1, q)$. The s -coordinate of the codeword vG is zero if and only if the point s is incident with the hyperplane π_v , as we saw in [Section 2.4](#).

We could also try and construct a multi-set S of points of $\text{PG}(k - 1, q)$ in which we can calculate (or at least bound) the size of the intersections of S with the hyperplanes of $\text{PG}(k - 1, q)$. Then [Lemma 4.15](#) implies that we can bound from below the minimum distance of the linear code we obtain from a generator matrix whose columns are vector representatives of the points of S .

Example 4.16

Let $\phi(X) = \phi(X_1, X_2, X_3)$ be an irreducible homogeneous polynomial over \mathbb{F}_q in three variables of degree m . Let S be the set of points of $\text{PG}(2, q)$ which are zeros of this

polynomial. Since ϕ is irreducible, each line of $\text{PG}(2, q)$ contains at most m points of S . By [Lemma 4.15](#), the matrix whose columns are a vector representative of the points of S is a $3 \times |S|$ matrix which generates a code with minimum distance at least $n - \deg \phi$. This can give an easy way to make codes with surprisingly good parameters. For example, suppose q is a square and we take the Hermitian curve, defined as the zeros of the polynomial

$$\phi(X) = X_1^{\sqrt{q}+1} + X_2^{\sqrt{q}+1} + X_3^{\sqrt{q}+1}.$$

This curve has $q\sqrt{q} + 1$ points and is irreducible. Thus we obtain a $[q\sqrt{q} + 1, 3, q\sqrt{q} - \sqrt{q}]_q$ code. ■

We say that two codes are **equivalent** if one can be obtained from the other by a permutation of the coordinates and permutations of the symbols in each coordinate. Note that non-linear codes can be equivalent to linear codes. Indeed, one can obtain a non-linear code (of the same size, length and minimum distance) from a linear code by simply permuting the symbols of \mathbb{F}_q in a fixed coordinate.

We can use S to obtain a model for all codes that are equivalent to a linear code C , this is called the **Alderson–Bruen–Silverman** model. Let S be the multi-set of n points of $\Sigma = \text{PG}(k - 1, q)$, obtained from the columns of a generator matrix G of the k -dimensional linear code C of length n . For each point $(s_1 : \dots : s_k)$ of S , we define a hyperplane π_s of $\Sigma = \text{PG}(k - 1, q)$ as the kernel of the linear form

$$\alpha_s(X) = s_1 X_1 + \dots + s_k X_k.$$

We embed Σ in a $\text{PG}(k, q)$ and consider $\text{PG}(k, q) \setminus \Sigma$ which, by [Exercise 2.12](#), is isomorphic to $\text{AG}(k, q)$. Within $\text{PG}(k, q)$, we label each hyperplane ($\neq \Sigma$) containing π_s with an element of \mathbb{F}_q . For each point v of the affine space $\text{PG}(k, q) \setminus \Sigma$ we obtain a codeword u of C' , a code equivalent to the code C . The coordinates of u are indexed by the elements of S , and the s -coordinate of u is the label given to the unique hyperplane of $\text{PG}(k, q)$ spanned by π_s and v . Observe that two codewords u and u' of C' (obtained from the points v and v' , respectively) agree in an s -coordinate if and only if $\alpha_s(v) = \alpha_s(v')$. The vectors vG and $v'G$ are codewords of C , so agree in at most $n - d$ coordinates, which implies that there are at most $n - d$ elements $s \in S$ such that $\alpha_s(v) = \alpha_s(v')$. Thus, u and u' agree in at most $n - d$ coordinates. Furthermore, there are two codewords which agree in exactly $n - d$ coordinates. Therefore, the code C' is of length n and minimum distance d . It is [Exercise 4.10](#), to prove that the code C' is equivalent to the linear code C . This model is used in [Exercise 4.11](#) to prove that if a linear code has a non-linear extension, then it has a linear extension.

4.5 Griesmer Bound

In [▶ Chapter 3](#) we proved various bounds involving the length, the minimum distance and the size of a block code. In this section, we shall prove another bound involving these

parameters, the Griesmer bound, which is specifically for linear codes. The Griesmer bound follows almost directly from the following lemma.

Lemma 4.17 *If there is a $[n, k, d]_q$ code, then there is a $[n - d, k - 1, \geq \lceil \frac{d}{q} \rceil]_q$ code.*

Proof

Let S be the multi-set of columns of a generator matrix G of a k -dimensional linear code C of length n and minimum distance d over \mathbb{F}_q .

By Lemma 4.15, there is a non-zero vector $v \in \mathbb{F}_q^k$ such that the hyperplane π_v of \mathbb{F}_q^k contains $n - d$ vectors of S . Let S' be this multi-set of $n - d$ vectors. Let G' be the $k \times (n - d)$ matrix whose columns are the vectors of S' . The matrix G' generates a linear code C' , obtained from G' by left multiplication by a vector of \mathbb{F}_q^k . The matrix G' is not, strictly speaking, a generator matrix of C' , since its rows are not linearly independent. The vector v is in the left nucleus of G' . The code C' is the subspace spanned by the rows of the matrix G' .

We want to prove that C' is a $(k - 1)$ -dimensional linear code. The rank of G' is at most $k - 1$, since $vG' = 0$. If the rank is less than $k - 1$, then there is another vector $v' \in \mathbb{F}_q^k$, not in the subspace spanned by v , for which $v'G' = 0$. But then we can find a $\lambda \in \mathbb{F}_q$ such that $(v + \lambda v')G$ has zeros in more than $n - d$ coordinates, which implies that C has non-zero codewords of weight less than d , which contradicts Lemma 4.1. Hence, C' is a $(k - 1)$ -dimensional linear code.

Let d' be the minimum distance of the code C' . By Lemma 4.15, there is a hyperplane π' of π_v which contains $n - d - d'$ vectors of S' . By Exercise 2.12, there are precisely $q + 1$ hyperplanes of \mathbb{F}_q^k containing the co-dimensional two subspace π' . Each one of these hyperplanes contains at most $n - d$ vectors of S and so at most d' vectors of $S \setminus \pi'$. Hence,

$$n \leq (q + 1)d' + n - d - d',$$

which gives

$$d' \geq \left\lceil \frac{d}{q} \right\rceil.$$

□

Theorem 4.18 (Griesmer bound)

If there is a $[n, k, d]_q$ code, then

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

Proof

By induction on k .

For $k = 1$ the bound gives $n \geq d$, which is clear.

By [Lemma 4.17](#), there is a $[n - d, k - 1, d']_q$ code, where

$$d' \geq \left\lceil \frac{d}{q} \right\rceil.$$

By induction,

$$n - d \geq \sum_{i=0}^{k-2} \left\lceil \frac{d'}{q^i} \right\rceil \geq \sum_{i=0}^{k-2} \left\lceil \frac{d}{q^{i+1}} \right\rceil = \sum_{i=1}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

□

Example 4.19

Consider the problem of determining the largest ternary code C of length 10 and minimum distance 4. The Plotkin bound from [Lemma 3.10](#) does not apply, since $d + n/r - n$ is negative. The sphere packing bound, [Theorem 3.9](#), implies

$$|C| \leq 3^{10}/21.$$

The Griesmer bound tells us that if there is a linear code with these parameters, then

$$10 \geq 4 + 2 + k - 2.$$

and so

$$|C| \leq 3^6.$$

To construct such a code, according to [Lemma 4.15](#), we need to find a set S of 10 points in $\text{PG}(5, 3)$ with the property that any hyperplane is incident with at most 6 points of S . Let G be the 6×10 matrix whose columns are vector representatives of the 10 points of S . The matrix G is the generator matrix of a $[10, 6, 4]_3$ code. Such a matrix G can be found directly, see [Exercise 4.14](#). However, we can construct such a code geometrically in the following way.

Let C^\perp be the linear code over \mathbb{F}_q generated by the $4 \times (q^2 + 1)$ matrix H , whose columns are the points of an elliptic quadric. For example, we could take the elliptic quadric defined as the zeros of the homogeneous quadratic form

$$X_1 X_2 - f(X_3, X_4),$$

where $f(X_3, X_4)$ is an irreducible homogeneous polynomial of degree two. Explicitly the points of the quadric are

$$\{(1, f(x, y), x, y) \mid x, y \in \mathbb{F}_q\} \cup \{(0, 1, 0, 0)\}.$$

As in the real projective space, the elliptic quadric has no more than two points incident with any line. To verify this algebraically, consider the line which is the intersection of the planes defined by $X_1 = a_3X_3 + a_4X_4$ and $X_2 = b_3X_3 + b_4X_4$. The x_3 and x_4 coordinates in the intersection with the quadric satisfy

$$(a_3x_3 + a_4x_4)(b_3x_3 + b_4x_4) - f(x_3, x_4) = 0,$$

which is a homogeneous polynomial equation of degree two in two variables. It is not identically zero, since f is irreducible, so there are at most two (projectively distinct or homogeneous) solutions for (x_3, x_4) ; the x_1 and x_2 coordinates are then determined by $x_1 = a_3x_3 + a_4x_4$ and $x_2 = b_3x_3 + b_4x_4$. This checks the intersection with q^4 lines, the intersection with the remaining lines can be checked similarly.

Therefore, any three columns of the matrix H are linearly independent, since three linearly dependent columns would imply three collinear points on the elliptic quadric. The elliptic quadric has four co-planar points, so H has four linearly dependent columns. By Lemma 4.4, C has a minimum distance 4 and is therefore a $[q^2 + 1, q^2 - 3, 4]_q$ code. Substituting $q = 3$, we obtain a ternary linear code C meeting the Griesmer bound.

The geometry also allows us to calculate the weight enumerator of C^\perp and hence the weight enumerator of C . Since any three points span a plane which intersects the elliptic quadric in a conic, and a conic contains $q + 1$ points, there are

$$\frac{(q^2 + 1)q^2(q^2 - 1)}{(q + 1)q(q - 1)} = (q^2 + 1)q$$

planes incident with $q + 1$ points of the elliptic quadric and the remaining $q^2 + 1$ planes are incident with exactly one point. This implies that C^\perp has $(q^2 + 1)q(q - 1)$ codewords of weight $q^2 - q$, $(q^2 + 1)(q - 1)$ codewords of weight q^2 and one codeword of weight zero.

For $q = 3$, the weight enumerator of C^\perp is

$$A^\perp(X) = 1 + 60X^6 + 20X^9.$$

The MacWilliams identities, Theorem 4.13, imply that C has weight enumerator,

$$A(X) = 1 + 60X^4 + 144X^5 + 60X^6 + 240X^7 + 180X^8 + 20X^9 + 24X^{10}.$$

Even if we do not restrict ourselves to linear codes, there is no larger code known with these parameters. The best known upper bound is $|C| \leq 891$. ■

Example 4.20

Consider the problem of determining if there is a $(16, 256, 6)_2$ code C , that is a binary code of length 16 with minimum distance 6 and size 256. The sphere packing bound, Theorem 3.9, implies

$$|C|(1 + 16 + \binom{16}{2}) \leq 2^{16},$$

which is satisfied. The Plotkin bound, [Theorem 3.12](#), does not give a contradiction since

$$|C| \leq d2^{n-2d+2} = 384.$$

Now, suppose that the code is linear, so C is a $[16, 8, 6]_2$ code. The Griesmer bound is also satisfied since,

$$n \geq 6 + \left\lceil \frac{6}{2} \right\rceil + \left\lceil \frac{6}{4} \right\rceil + \sum_{i=3}^7 \left\lceil \frac{6}{2^i} \right\rceil = 16.$$

However, [Lemma 4.17](#) implies the existence of a $[10, 7, \geq 3]_2$ code. This code is a 1-error correcting binary code of length 10, so the sphere packing bound, [Theorem 3.9](#), implies that

$$(1 + 10)2^7 \leq 2^{10},$$

which is a contradiction. Therefore, there is no $[10, 7, \geq 3]_2$ code. Hence, there is no $[16, 8, 6]_2$ code. However, there is a non-linear $(16, 256, 6)_2$ code and we shall construct one both in [Chapter 9](#) and in [Chapter 10](#). ■

4.6 Constructing Designs from Linear Codes

A τ -design is a collection \mathcal{D} of κ -subsets of $\{1, \dots, n\}$ with the property that every τ -subset of $\{1, \dots, n\}$ is contained in precisely λ subsets of \mathcal{D} , for some fixed positive integer λ . If we want to specify the parameters, then we say that \mathcal{D} is a τ - (n, κ, λ) design.

Let $u \in \mathbb{F}_q^n$. The **support** of $u = (u_1, \dots, u_n)$ is a subset of $\{1, \dots, n\}$ defined as

$$\{i \in \{1, \dots, n\} \mid u_i \neq 0\}.$$

In this section we shall prove that if the codewords of the dual of a linear code C have few distinct weights, then one can construct τ -designs from the supports of codewords of C of a fixed weight. Before proving the main theorem, we will prove by counting that we can construct a 3-design from the extended Hamming code, [Example 4.11](#).

Example 4.21

In [Example 4.14](#), we calculated the weight distribution for the extended Hamming code in [Example 4.11](#) and deduced that there are 14 codewords of weight 4. Two codewords u and v of weight 4 have at most two 1's in common, since otherwise $u + v$ would be a codeword of weight 2. Therefore, every 3-subset of $\{1, \dots, 8\}$ is contained in the support of at most one codeword of weight 4. There are $14 \binom{4}{3} = 56$ subsets of size 3 of the 14 supports of the 14 codewords of weight 4 and $\binom{8}{3} = 56$ subsets of size 3 of $\{1, \dots, 8\}$. Hence, each 3-subset is

contained in a unique support of a codeword of weight 4 and we have deduced that the set of these supports is a 3-(8, 4, 1) design. ■

In the following theorem, κ can be any number in the set $\{d, \dots, n\}$ in the case that $q = 2$, since the condition is vacuous. If $q \neq 2$, then, by [Exercise 4.15](#), the condition is surely satisfied if

$$\kappa \in \{d, \dots, d - 1 + \left\lfloor \frac{d - 1}{q - 2} \right\rfloor\}.$$

In order to simplify the statement of the following theorem, we say that C has a weight w if there is a codeword of C of weight w .

Theorem 4.22

Let C be an $[n, k, d]_q$ code such that C^\perp has at most $d - \tau$ non-zero weights of weight at most $n - \tau$, for some $\tau \leq d - 1$. If κ has the property that two codewords of C of weight κ have the same support if and only if they are multiples of each other, then the set of supports of the codewords of C of weight κ is a τ -(n, κ, λ) design, for some λ .

Proof

Let T be a τ -subset of $\{1, \dots, n\}$. Let $C \setminus T$ be the code obtained from C by deleting the coordinates indicated by the elements of T . If after deleting τ coordinates the codewords u and v are the same, then u and v differ in at most τ coordinates. Since $\tau \leq d - 1$, this cannot occur, so deleting the coordinates does not reduce the number of codewords. Hence, $C \setminus T$ is a k -dimensional linear code of length $n - \tau$.

Let C_T^\perp be the subset of codewords of C^\perp which have zeros in all the coordinates indicated by the elements of T . Then $C_T^\perp \setminus T$ is a linear code and

$$C_T^\perp \setminus T \subseteq (C \setminus T)^\perp,$$

since a vector in C_T^\perp is orthogonal to all the vectors of C and has zeros in the coordinates indicated by the elements of T . Furthermore,

$$\dim(C_T^\perp \setminus T) = \dim C_T^\perp$$

since the codewords of C_T^\perp have zeros in the coordinates indexed by T , so deleting these coordinates does not reduce the number of codewords.

Let H be a generator matrix for C^\perp . Let L be the set of τ vectors of \mathbb{F}_q^{n-k} which are the columns of H indicated by the elements of T . Then

$$C_T^\perp = \{vH \mid v \in \mathbb{F}_q^{n-k}, v \cdot s = 0, \text{ for all } s \in L\},$$

since vH is a codeword of C^\perp and has zeros in the coordinates indexed by T precisely when $v \cdot s = 0$, for all $s \in L$.

Hence,

$$\dim C_T^\perp \geq n - k - \tau.$$

Now,

$$\dim(C \setminus T) = k$$

implies

$$\dim(C \setminus T)^\perp = n - \tau - k$$

and we just proved that

$$\dim(C_T^\perp \setminus T) \geq n - \tau - k,$$

so we have that

$$C_T^\perp \setminus T = (C \setminus T)^\perp.$$

The weight of a codeword of $C_T^\perp \setminus T$ is the weight of the corresponding codeword of C^\perp . By hypothesis, C^\perp has at most $d - \tau$ non-zero weights of weight at most $n - \tau$. Since at least τ of the coordinates of a codeword of C_T^\perp are zero, C_T^\perp has weights at most $n - \tau$. Therefore, $(C \setminus T)^\perp$ has at most $d - \tau$ non-zero weights.

Since $C \setminus T$ has minimum distance at least $d - \tau$, [Exercise 4.16](#) implies that the weight enumerator of $C \setminus T$ is determined.

If u is a non-zero codeword, then μu is another codeword with the same support as u , for all non-zero $\mu \in \mathbb{F}_q$. The number $\lambda(q - 1)$, of codewords of $C \setminus T$ of weight $\kappa - \tau$, is determined by the weight enumerator of $C \setminus T$. The number λ does not depend on which subset T we choose, only the size of the subset T . By induction on κ , for all τ -subsets T of $\{1, \dots, n\}$, there are a fixed number of supports of the codewords of weight κ containing T . Therefore, the set of the supports of the codewords of C of weight κ is a τ - (n, κ, λ) design. \square

Example 4.23

Consider the $[10, 6, 4]_3$ code from [Example 4.19](#). The dual code C^\perp has codewords of weight 0, 6 and 9 so, according to [Theorem 4.22](#), the set of supports of the codewords of weight κ is a 3-design, provided that no two codewords of C of weight κ have the same support. By [Exercise 4.15](#), we can be assured of this for $\kappa \in \{4, 5, 6, 7\}$.

To calculate λ , we count in two ways the number of 3-subsets. Each 3-subset of $\{1, \dots, 10\}$ is contained in λ 3-subsets of the design, so

$$\binom{10}{3} \lambda = \binom{\kappa}{3} \alpha,$$

where α is the number of supports of codewords of C of weight κ . The number of supports of codewords of weight κ is the number of codewords of weight κ divided by $q - 1$.

Therefore, from the code C we can construct a 3-(10, 4, 1)-design, a 3-(10, 5, 6)-design and a 3-(10, 6, 5)-design. ■

In [Example 4.23](#), we could have constructed the designs directly from the elliptic quadric. For example, the 3-(10, 4, 1) design is obtained by taking subsets of 4 coplanar points and the 3-(10, 5, 6) design is obtained by taking subsets of 5 points, no 4 coplanar. In [Chapter 5](#) we shall construct codes from polynomial divisors of $X^n - 1$ which will often satisfy the hypothesis of [Theorem 4.22](#) and allow us to construct designs. In many cases, these designs cannot be constructed directly from any geometrical object.

4.7 Comments

The MacWilliams identities from [Chapter 4](#) appear in MacWilliams' thesis "Combinatorial Problems of Elementary Group Theory", although the standard reference is [\[50\]](#). The MacWilliams identities lead to a set of constraints on the existence of an $[n, k, d]_q$ code. We have that $A_0 = 1$ and $A_1 = \dots = A_{d-1} = 0$ and that

$$1 + A_d + \dots + A_n = q^k.$$

Since

$$A_i^\perp \geq 0,$$

[Theorem 4.13](#) implies, for a fixed n and q , the linear constraint

$$\sum_{j=0}^n A_j K_i(j) \geq 0.$$

The coefficients

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

are called the **Krawtchouk polynomials**. Delsarte [\[21\]](#) proved that from the distance distribution between the codewords of an arbitrary code (not necessarily a linear code) one can deduce similar inequalities, called the **linear programming bound**. This can be a powerful tool, not only in ruling out certain parameter sets, but also for the construction of codes, since it can give significant information about the distance distribution.

The Griesmer bound is from [31] and the Hamming code was first considered by Hamming in [34]. The upper bound on the size of the code in Example 4.19 is from [55].

The Alderson–Bruen–Silverman model for codes equivalent to linear codes in ▶ Section 4.4 is from [2]. The fact that a linear code with a non-linear extension has a linear extension, Exercise 4.11, is due to Alderson and Gács, see [1].

Theorem 4.22 is the Assmus–Mattson theorem from [4].

The bound in Exercise 4.3 is due to Varshamov [75] and is known as the linear Gilbert–Varshamov bound.

4.8 Exercises

4.1 Prove that if C is linear, then the extended code \overline{C} is linear.

4.2 Prove that the code in Example 4.2 is a perfect code.

4.3 Prove that if

$$\sum_{j=0}^{d-2} \binom{n-1}{j} (q-1)^j < q^{n-k},$$

then there exists an $[n, k, d]_q$ code.

4.4 Prove that the system of equations in Example 4.14 has the solution

$$A(X) = 1 + 14X^4 + X^8 + \lambda(X^2 - 2X^4 + X^6).$$

4.5 Prove that the code in Example 4.8 has minimum distance 4 and decode the received vector $(0, 1, 1, 0, 2, 2, 2, 0)$ using syndrome decoding.

4.6 Prove that the code C in Example 3.4 is linear but not self-dual although for the weight enumerator $A(X)$ of C , we have $A(X) = A^\perp(X)$. Prove that C is equivalent to C^\perp .

4.7 Let C be the linear code over \mathbb{F}_5 generated by the matrix

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

Calculate the minimum distance of C and decode the received vector $(0, 2, 3, 4, 3, 2)$ using syndrome decoding.

4.8 Let C be the linear code over \mathbb{F}_7 defined by the check matrix

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 1 & 4 & 2 & 2 & 4 & 1 \\ 0 & 1 & 1 & 6 & 1 & 6 & 6 \end{pmatrix}.$$

- i. Prove that C is a $[7, 3, 5]_7$ code.
- ii. Decode the received vector $(2, 2, 3, 6, 1, 2, 2)$ using syndrome decoding.

4.9 Let C be the 3-dimensional linear code over \mathbb{F}_3 generated by the matrix

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

Prove that C has minimum distance 6 and use syndrome decoding to decode the received vector

$$(1, 2, 0, 2, 0, 2, 0, 0, 0).$$

4.10 Prove that the code C' obtained from the Alderson–Bruen–Silverman model is equivalent to the linear code C from which the model is set up.

4.11 Let S be the set of n vectors obtained from the set of columns of a generator matrix of a linear code C and suppose that C has an extension to a code of length $n + 1$ and minimum distance $d + 1$.

- i. Prove that there is a function

$$f : \mathbb{F}_q^k \rightarrow \mathbb{F}_q$$

with the property that if $f(u) = f(v)$, then $u - v$ is orthogonal (with respect to the standard inner product) to less than $n - d$ points of S .

- ii. Let T be the set of vectors of \mathbb{F}_q^k which are orthogonal to $n - d$ vectors of S . Let $v \in T$ and let u_1, \dots, u_{k-2} be a set of $k - 2$ vectors extending v to a set of $k - 1$ linearly independent vectors. Prove that for all $\lambda_1, \dots, \lambda_{k-2}, \lambda, \mu \in \mathbb{F}_q, \lambda \neq \mu$,

$$f(\lambda_1 u_1 + \dots + \lambda_{k-2} u_{k-2} + \lambda v) \neq f(\lambda_1 u_1 + \dots + \lambda_{k-2} u_{k-2} + \mu v).$$

- iii. Prove that if every hyperplane of \mathbb{F}_q^k contains a vector of T , then every hyperplane of \mathbb{F}_q^k contains q^{k-2} vectors u such that $f(u) = 0$.
- iv. Prove that there is a hyperplane of \mathbb{F}_q^k not containing a vector of T .
- v. Prove that C has a linear extension. In other words, it can be extended to a $[n+1, k, d+1]_q$ code.

4.12 Prove that for fixed $r = n - d$, the Griesmer bound implies $n \leq (r - k + 2)q + r$.

4.13 Let $r = n - d$ and let S be the set of columns of a generator matrix of a 3-dimensional linear code C of length $(r - 1)q + r$, so we have equality in the bound of [Exercise 4.12](#). Prove that S is a set of vectors of \mathbb{F}_q^k in which every hyperplane contains 0 or r vectors of S . Equivalently show that the non-zero codewords of C have weight n or d .

4.14

- i. Verify that equality in the Griesmer bound occurs for the parameters of the code C in [Example 4.19](#) if and only if $q = 3$.
- ii. Let G be a 6×10 matrix

$$G = \left(I_6 \mid A \right).$$

Let S be the set of rows of the 6×4 matrix A , considered as 6 points of $\text{PG}(3, 3)$. Prove that G is a generator matrix of a $[10, 6, 4]_3$ code if and only if S has the property all points of S have weight at least three (i.e. the points of S have at most one zero coordinate), no two points of S are collinear with a point of weight one and that no three points of S are collinear.

- iii. Find a matrix A so that G is a generator matrix for a $[10, 6, 4]_3$ code.

4.15 Let C be a linear code over \mathbb{F}_q , where $q \neq 2$.

- i. Prove that if $w - \lceil w/(q - 1) \rceil < d$, where d is the minimum distance of a linear code C , then two codewords of C of weight w have the same support if and only if they are multiples of each other.
- ii. Prove that if $w \leq (d - 1)(q - 1)/(q - 2)$, then $w - \lceil w/(q - 1) \rceil < d$.

4.16 Let C be a linear code of length n and minimum distance d with the property that C^\perp has at most d distinct weights, w_1, \dots, w_d .

- i. Let A_j denote the number of codewords of C of weight j and let A_j^\perp denote the number of codewords of C^\perp of weight j . Prove that

$$q^k \sum_{j=0}^n A_j^\perp (1 - X)^j = (1 + (q - 1)(1 - X))^n + \sum_{j=d}^n A_j X^j (1 + (q - 1)(1 - X)^{n-j}).$$

- ii. Prove that the $n + 1$ polynomials $X^{n-r}(1 + (q - 1)(1 - X)^r)$ ($r = 0, \dots, n - d$), $(1 - X)^{w_j}$ ($j = 1, \dots, d$) are linearly independent.
- iii. Prove that the weight enumerator of C^\perp is determined.
- iv. Prove that the weight enumerator of C is determined.



Cyclic Codes

Although it will turn out that cyclic codes are not asymptotically good codes, they are an important class of codes which include many useful and widely implemented short length codes, most notably the Golay codes and the general class of BCH codes. BCH codes have a prescribed minimum distance which means that, by construction, we can bound from below the minimum distance and therefore guarantee some error-correction properties. Cyclic codes also provide examples of linear codes with few weights, which allows us to construct designs via [Theorem 4.22](#). The cyclic structure of these codes will appear again in [Chapter 10](#), when we consider p -adic codes.

5.1 Basic Properties

A linear code C is called **cyclic** if, for all $(c_1, \dots, c_n) \in C$, the vector $(c_n, c_1, \dots, c_{n-1}) \in C$.

The map

$$(c_1, \dots, c_n) \mapsto c_1 + c_2X + \dots + c_nX^{n-1}$$

is a bijection between the vectors of \mathbb{F}_q^n and the polynomials in

$$\mathbb{F}_q[X]/(X^n - 1).$$

We define the **weight** $\text{wt}(u)$ of a polynomial $u(X) \in \mathbb{F}_q[X]/(X^n - 1)$ of degree less than n , as the weight of the corresponding vector of \mathbb{F}_q^n . In other words, the number of non-zero coefficients that it has.

An **ideal** I of a polynomial ring is a subspace with the property that if $f \in I$, then $Xf \in I$.

Lemma 5.1 A cyclic code C is mapped by the bijection to an ideal I in $\mathbb{F}_q[X]/(X^n - 1)$.

Proof

This is precisely the condition that a linear code satisfies to be cyclic. \square

We assume that $(n, q) = 1$ so that the polynomial $X^n - 1$ has no repeated factors in its factorisation, see ▶ Section 2.3.

The ring $\mathbb{F}_q[X]/(X^n - 1)$ is a principal ideal ring, so I in Lemma 5.1 is a principal ideal. Hence,

$$I = \langle g \rangle = \{fg \mid f \in \mathbb{F}_q[X]/(X^n - 1)\}$$

for some polynomial g , which is monic and of lowest degree in the ideal.

Therefore, a cyclic code C is mapped by the bijection to $\langle g \rangle$. We will from now on write $C = \langle g \rangle$, for some polynomial g .

Lemma 5.2 If $C = \langle g \rangle$ is a cyclic code of length n , then g divides $X^n - 1$ and C has dimension at least $n - \deg g$.

Proof

If $g(X)$ does not divide $X^n - 1$, then, using the Euclidean algorithm, we can find polynomials $a(X)$ and $b(X)$ such that

$$a(X)g(X) + b(X)(X^n - 1)$$

is equal to the greatest common divisor of $g(X)$ and $X^n - 1$, which has degree less than g . This contradicts the property that g has minimal degree in the ideal I . Therefore, g divides $X^n - 1$.

The polynomials $X^j g$, for $j = 0, \dots, n - \deg(g) - 1$ are linearly independent polynomials in $\langle g \rangle$, so the dimension of C is at least $n - \deg g$. \square

In fact, we shall see that the dimension k of C is precisely $n - \deg g$. This follows from the following theorem.

Theorem 5.3

Let $C = \langle g \rangle$ be a cyclic code of length n . The dual code C^\perp is the cyclic code $\langle \overleftarrow{h} \rangle$, where $g(X)h(X) = X^n - 1$ and $\overleftarrow{h}(X) = X^k h(X^{-1})$.

Proof

Suppose that

$$g(X) = \sum_{j=0}^{n-k} g_j X^j$$

and

$$h(X) = \sum_{i=0}^k h_i X^i.$$

The code $\langle g \rangle$ contains the row span of the $k \times n$ matrix

$$G = \begin{pmatrix} g_0 & \dots & g_{n-k} & 0 & \dots & \dots & 0 \\ 0 & g_0 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & 0 \\ 0 & \dots & \dots & 0 & g_0 & \dots & g_{n-k} \end{pmatrix}$$

and the code $\langle \overleftarrow{h} \rangle$ contains the row span of the $(n-k) \times n$ matrix

$$H = \begin{pmatrix} h_k & \dots & h_0 & 0 & \dots & \dots & 0 \\ 0 & h_k & \dots & h_0 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & 0 \\ 0 & \dots & \dots & 0 & h_k & \dots & h_0 \end{pmatrix}.$$

The scalar product between the s -th row of G and the r -th row of H , where $s \in \{1, \dots, k\}$ and $r \in \{1, \dots, n-k\}$ is

$$\sum_{i=s}^{k+r} g_{i-s} h_{k+r-i},$$

which is the coefficient of X^{k+r-s} in gh . Since $1 \leq k+r-s \leq n-1$, this coefficient is zero and so $GH^t = 0$.

Since

$$n = \dim C + \dim C^\perp \geq \text{rank}(G) + \text{rank}(H) = n, \quad (5.1)$$

the theorem follows. \square

Corollary 5.4 *The code $C = \langle g \rangle$ of length n has dimension $n - \deg g$.*

Proof

Let G and H be as in the previous proof. Equation (5.1) implies that the dimension of C is the rank of G , which is k . \square

Example 5.5 (perfect ternary Golay code)

Consider the factorisation of $X^{11} - 1$ over \mathbb{F}_3 . As in ► Section 2.3, we calculate the cyclotomic subsets of the multiples of 3 modulo 11,

$$\{0\}, \{1, 3, 9, 5, 4\}, \{2, 6, 7, 10, 8\}.$$

According to Lemma 2.12, there are two factors of degree 5 which are

$$(X - \alpha)(X - \alpha^3)(X - \alpha^9)(X - \alpha^5)(X - \alpha^4)$$

and

$$(X - \alpha^2)(X - \alpha^6)(X - \alpha^7)(X - \alpha^{10})(X - \alpha^8),$$

where α is a primitive 11-th root of unity in \mathbb{F}_{3^5} .

Suppose that

$$X^5 + a_4X^4 + a_3X^3 + a_2X^2 + a_1X + a_0$$

is the first of these factors. Then $a_0 = -\alpha^{22} = -1$. Since the roots of the first factor are the reciprocals of the roots of the second factor, the second factor is

$$X^5 - a_1X^4 - a_2X^3 - a_3X^2 - a_4X - 1.$$

It is fairly easy to deduce from this that the factorisation is

$$X^{11} - 1 = (X - 1)(X^5 - X^3 + X^2 - X - 1)(X^5 + X^4 - X^3 + X^2 - 1).$$

The cyclic code $C = \langle X^5 - X^3 + X^2 - X - 1 \rangle$ over \mathbb{F}_3 is the perfect ternary Golay code of length 11. To prove that this is a perfect code we need to show that the minimum weight of a non-zero codeword is 5 (and hence the minimum distance is 5 according to Lemma 4.1) and observe that

$$\left(1 + 2\binom{11}{1} + 4\binom{11}{2}\right)3^6 = 3^{11},$$

so the sphere-packing bound of Theorem 3.9 is attained.

Adding a column of 1's to the generator matrix

$$\begin{pmatrix} -1 & -1 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 0 & 1 \end{pmatrix}$$

we get a generator matrix of a self-dual code \overline{C} of length 12. This we can check by computing the scalar product of any two rows and verifying that it is zero (modulo 3). Since this code is self-dual, the codewords have weights which are multiples of 3. If we can rule out the possibility that a codeword has weight 3, which we will in ▶ Section 5.3, then the minimum weight of a non-zero codeword of \overline{C} is 6, which implies that the minimum weight of a non-zero codeword of the cyclic code $\langle X^5 - X^3 + X^2 - X - 1 \rangle$ is 5. Therefore, C is a $[[11, 6, 5]]_3$ code and \overline{C} is a $[[12, 6, 6]]_3$ code. ■

5.2 Quadratic Residue Codes

Let n and q be primes for which q is a square in \mathbb{F}_n , where we consider the field $\mathbb{F}_n \cong \mathbb{Z}/n\mathbb{Z}$ to be addition and multiplication modulo n , defined on the set $\{0, 1, \dots, n-1\}$.

Let α be a primitive n -th root of unity in some extension field of \mathbb{F}_q .

Define

$$g(X) = \prod (X - \alpha^r),$$

where the product runs over the non-zero squares r in \mathbb{F}_n .

Lemma 5.6 *The polynomial $g(X)$ divides $X^n - 1$ in $\mathbb{F}_q[X]$.*

Proof

Since q is a square in \mathbb{F}_n , the map

$$r \mapsto qr$$

is a bijection from the squares of \mathbb{F}_n to the squares of \mathbb{F}_n , for all non-zero squares $r \in \mathbb{F}_n$.

Hence,

$$g(X) = \prod (X - \alpha^r) = \prod (X - \alpha^{rq}),$$

where the product runs over the non-zero squares r in \mathbb{F}_n .

Lemma 2.11 implies that $g(X) \in \mathbb{F}_q[X]$ and note that the roots of $g(X)$ are distinct n -th roots of 1. □

Since $g(X)$ is a factor of $X^n - 1$, we can define the cyclic code $\langle g \rangle$ of length n over \mathbb{F}_q . This code is called the **quadratic residue code**.

We can obtain evidence that the minimum distance of a quadratic residue code is quite good from the following theorems.

Theorem 5.7

If $u \in \langle g \rangle$ and $u(1) \neq 0$, then $\text{wt}(u)^2 \geq n$.

Proof

Since $u \in \langle g \rangle$, the n -th roots of unity α^r of \mathbb{F}_q , where r is a non-zero square in \mathbb{F}_n , are zeros of $u(X)$.

Let t be a non-square of \mathbb{F}_n . The n -th roots of unity α^s of \mathbb{F}_q , where s is a non-square in \mathbb{F}_n , are zeros of $u(X^t)$, since the product of two non-squares is a square. Therefore, all the n -th roots of unity of \mathbb{F}_q , except 1, are zeros of $u(X)u(X^t)$. Hence,

$$u(X)u(X^t) = (1 + X + \cdots + X^{n-1})v(X),$$

for some polynomial $v(X)$. Since $u(1) \neq 0$, we have that $v(1) \neq 0$.

Therefore, in the ring $\mathbb{F}_q[X]/(X^n - 1)$,

$$u(X)u(X^t) = (1 + X + \cdots + X^{n-1})v(1),$$

since $v(X) = v(1) + (X - 1)v_1(X)$, for some polynomial $v_1(X)$.

Since $u(X)$ has $\text{wt}(u)$ terms, this implies that $\text{wt}(u)^2 \geq n$. □

Theorem 5.8

If $n \equiv -1 \pmod{4}$, $u \in \langle g \rangle$ and $u(1) \neq 0$, then $\text{wt}(u)^2 - \text{wt}(u) + 1 \geq n$.

Proof

If $n \equiv -1 \pmod{4}$, then -1 is a non-square in \mathbb{F}_n , since $(-1)^{(n-1)/2} = -1$. Therefore, in the proof of [Theorem 5.7](#), we can take $t = -1$. Then,

$$u(X)u(X^{-1}) = (1 + X + \cdots + X^{n-1})v(1).$$

In the product there are at least $\text{wt}(u)$ terms of $u(X)$ which multiply with a term of $u(X^{-1})$ to give a constant term, since $X^j X^{-j} = 1$. Hence,

$$\text{wt}(u)^2 - \text{wt}(u) \geq n - 1.$$

□

Example 5.9 (perfect binary Golay code)

Consider the quadratic residue code with $n = 23$ and $q = 2$. Let ϵ be a primitive element of $\mathbb{F}_{2^{11}} \cong \mathbb{F}_2[X]/(X^{11} + X^2 + 1)$ and let $\alpha = \epsilon^{89}$. Then α is a primitive 23-rd root of unity. By [Lemma 5.6](#), the factorisation of $X^{23} - 1$ in $\mathbb{F}_2[X]$ has a factor

$$g(X) = \prod_{r \in S} (X - \alpha^r),$$

where $S = \{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$ is the set of non-zero squares of \mathbb{F}_{23} .

5.3 BCH Codes

Let α be a primitive n -th root of unity in \mathbb{F}_{q^m} . BCH codes are a class of cyclic codes in which we choose α so that $\alpha, \alpha^2, \dots, \alpha^{d_0-1}$ are roots of a low degree polynomial g of $\mathbb{F}_q[X]$, for some $d_0 < n$. This allows us to bound the minimum distance of the code $\langle g \rangle$. The lower the degree of g , the larger the dimension (and hence the size) of the code.

Suppose that $g(X) \in \mathbb{F}_q[X]$ is the polynomial of minimal degree such that

$$g(\alpha^j) = 0,$$

for $j = 1, \dots, d_0 - 1$.

The code $\langle g \rangle$ is called a **BCH code**, after Bose, Ray-Chaudhuri and Hocquenghem who introduced this family of cyclic codes. The parameter d_0 is called the **prescribed minimum distance** because of the following theorem.

Theorem 5.10

The dimension of the BCH code $\langle g \rangle$ is at least $n - m(d_0 - 1)$ and its minimum distance is at least d_0 .

Proof

Let $j \in \{1, \dots, d_0 - 1\}$. By [Lemma 2.11](#), the polynomial

$$(X - \alpha^j)(X - \alpha^{jq}) \cdots (X - \alpha^{jq^{m-1}})$$

is in $\mathbb{F}_q[X]$. Clearly, it is zero at α^j . Since this polynomial has degree m this implies that there is a polynomial of degree $m(d_0 - 1)$ in $\mathbb{F}_q[X]$ which is zero at α_j , for all $j = 1, \dots, d_0 - 1$.

Thus, the degree of g is at most $m(d_0 - 1)$ so, by [Corollary 5.4](#), the dimension of $\langle g \rangle$ is at least $n - m(d_0 - 1)$.

Suppose that there is an $f \in \langle g \rangle$ for which $\text{wt}(f)$ is at most $d_0 - 1$. Then

$$f(X) = b_1 X^{k_1} + \cdots + b_{d_0-1} X^{k_{d_0-1}},$$

for some k_1, \dots, k_{d_0-1} .

Since $f \in \langle g \rangle$,

$$f(\alpha^j) = 0$$

for all $j = 1, \dots, d_0 - 1$. Writing this in matrix form these equations are

$$\begin{pmatrix} \alpha^{k_1} & \alpha^{k_2} & \cdots & \alpha^{k_{d_0-1}} \\ \alpha^{2k_1} & \alpha^{2k_2} & \cdots & \alpha^{2k_{d_0-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(d_0-1)k_1} & \alpha^{(d_0-1)k_2} & \cdots & \alpha^{(d_0-1)k_{d_0-1}} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{d_0-1} \end{pmatrix} = 0.$$

The determinant of the matrix is

$$\prod_{i \neq j} (\alpha^{k_i} - \alpha^{k_j}),$$

which is non-zero. This implies that the only solution to the above system is $f(X) = 0$. Hence, the minimum weight of a non-zero codeword of the cyclic code $\langle g \rangle$ is at least d_0 . The lemma follows since, by [Lemma 4.1](#), the minimum weight of a non-zero codeword of a linear code is equal to its minimum distance. \square

Example 5.11

Let α be a primitive 31-st root of unity in \mathbb{F}_{32} . By [Lemma 2.12](#), we obtain the factorisation of $X^{31} - 1$ over \mathbb{F}_2 by considering the cyclotomy classes

$$\begin{aligned} &\{1, 2, 4, 8, 16\}, \{3, 6, 12, 24, 17\}, \{5, 10, 20, 9, 18\}, \{7, 14, 28, 25, 19\}, \\ &\{11, 22, 13, 26, 21\}. \end{aligned}$$

The i -th cyclotomy class gives a polynomial $f_i(X)$ in $\mathbb{F}_2[X]$ which is zero at α^j for j in the cyclotomy class. For example,

$$f_1(X) = (X - \alpha)(X - \alpha^2)(X - \alpha^4)(X - \alpha^8)(X - \alpha^{16})$$

is in $\mathbb{F}_2[X]$ and is zero at α^j for $j \in \{1, 2, 4, 8, 16\}$.

Let

$$g(X) = f_1(X)f_2(X)f_3(X).$$

According to [Corollary 5.4](#), the cyclic code $\langle g \rangle$ is a 16-dimensional linear code.

Since 1, 2, 3, 4, 5 and 6 appear in the first three cyclotomic subsets,

$$g(\alpha^j) = 0,$$

for $j = 1, \dots, 6$. [Theorem 5.10](#) implies that $\langle g \rangle$ is a $[31, 16, \geq 7]_2$ code. It is in fact a $[31, 16, 7]_2$ code. Since there exists a $[31, 16, 8]_2$ code, $\langle g \rangle$ is not an optimal linear code for this length and dimension. \blacksquare

Example 5.12 (shortened Reed–Solomon code)

Let α be a primitive $(q - 1)$ -st root of unity in \mathbb{F}_q . By [Theorem 2.4](#), the polynomial $X^{q-1} - 1$ factorises into linear factors over \mathbb{F}_q . Each cyclotomy class has size 1 and the factors are

$$f_i(X) = X - \alpha^i,$$

for $i = 0, \dots, q - 2$.

Let

$$g(X) = f_1(X)f_2(X)\cdots f_{d-1}(X).$$

According to [Corollary 5.4](#), $\langle g \rangle$ is a $(n - d + 1)$ -dimensional linear code of length n . According to [Theorem 5.10](#), $\langle g \rangle$ has minimum distance at least d . This is an example of an MDS code, which we will study in more depth in [Chapter 6](#). ■

Example 5.13

In [Example 5.9](#), the numbers 1, 2, 3 and 4 appear in the same cyclotomy class, so [Theorem 5.10](#) implies that the binary Golay code has weight at least 5. As observed in [Example 5.9](#), this implies that the extended binary Golay code \bar{C} has no codewords of weight 4, which implies that the minimum distance of \bar{C} is 8. This, in turn, implies that the minimum distance of the binary Golay code is 7. ■

Example 5.14

[Theorem 5.10](#) generalises in a straightforward way to [Exercise 5.5](#). We can now establish that the minimum distance of the ternary Golay code is 5. By [Exercise 5.5](#), since 3, 4 and 5 appear in the same cyclotomy class (and 6, 7 and 8 appear in the same cyclotomy class), the ternary Golay code in [Example 5.5](#) has minimum distance at least 4. Therefore, the extended code \bar{C} has no codewords of weight three, so the weight of a non-zero codeword of the extended code is either 6, 9 or 12. As observed in [Example 5.5](#), this implies that the minimum distance of the ternary Golay code is 5. ■

The following theorem, which we quote without proof, states that there is no sequence of asymptotically good BCH codes.

Theorem 5.15

There is no infinite sequence of $[n, k, d]_q$ BCH codes for which both $\delta = d/n$ and $R = k/n$ are bounded away from zero.

5.4 Comments

The introduction of cyclic codes and quadratic residue codes is widely accredited to Eugene Prange and Andrew Gleason who proved the automorphism group of an extended quadratic residue code has a subgroup which is isomorphic to either $\text{PSL}(2, p)$ or $\text{SL}(2, p)$, see [\[12\]](#). The Golay codes were discovered by Golay [\[27\]](#). The BCH codes were introduced by Bose and Ray-Chaudhuri in [\[13\]](#) and independently by Hocquenghem in [\[38\]](#). The fact that long BCH codes are asymptotically bad is proven by Lin and Welden in [\[47\]](#). The code in [Exercise 5.7](#) is a Zetterberg code, one of a family of $[4^m + 1, 4^m + 1 - 4m, 5]_2$ codes.

5.5 Exercises

5.1 Let \overline{C} be the extended ternary Golay code from [Example 5.5](#).

- i. Verify that the factorisation of $X^{11} - 1$ in $\mathbb{F}_3[X]$ is as in [Example 5.5](#).
- ii. Prove that the weight enumerator of \overline{C} is

$$A(X) = 1 + 264X^6 + 440X^9 + 24X^{12}.$$

- iii. Let S be the set of 12 points of $\text{PG}(5, 3)$ obtained from the set of columns of a generator matrix of the code \overline{C} . Label the points of S by the elements of $\{1, \dots, 12\}$ and define a set D of 6-subsets to be the points of S which are dependent (i.e. are contained in a hyperplane of $\text{PG}(5, 3)$). Prove that D is a 5-(12, 6, 1) design.
- iv. Verify that [Theorem 4.22](#) implies that the set of supports of the codewords of weight 6 of \overline{C} is a 5-(12, 6, 1) design.

5.2 Prove that in [Example 5.9](#) the code $\langle \overleftarrow{g} \rangle$ is equivalent to the code $\langle g \rangle$.

5.3

- i. Prove that the extended Golay code over \mathbb{F}_2 , the code \overline{C} in [Example 5.9](#), is self-dual and that the weights of the codewords of \overline{C} are multiples of 4.
- ii. Prove that the weight enumerator of the code \overline{C} is

$$A(X) = 1 + 759X^8 + 2576X^{12} + 759X^{16} + X^{24}.$$

- iii. Apply [Theorem 4.22](#) to construct a 5-(24, 8, 1) design.

5.4 Investigate the observation that if $n \equiv -1$ modulo 4 and $\langle g \rangle$ is a quadratic residue code, then the reverse of the polynomial $(X^n - 1)/(X - 1)g(X)$ is $g(X)$. Does this imply that the extension of the code $\langle g \rangle$ is self-dual?

5.5 Suppose that $g(X) \in \mathbb{F}_q[X]$ is the polynomial of minimal degree such that

$$g(\alpha^j) = 0,$$

for $j = \ell + 1, \dots, \ell + d_0 - 1$.

Prove that the dimension of $\langle g \rangle$ is at least $n - m(d_0 - 1)$ and the minimum distance of $\langle g \rangle$ is at least d_0 .

5.6 Construct the largest possible BCH code with the following parameters.

- i. A binary code of length 15 with minimum distance at least 5.
- ii. A binary code of length 31 with minimum distance at least 11.
- iii. A ternary code of length 13 with minimum distance at least 7.

Compare the dimension of the codes with the Griesmer bound, the sphere-packing bound and the Gilbert–Varshamov bound.

5.7

- i. Prove that $X^{17} + 1$ factorises in $\mathbb{F}_2[X]$ as $(X + 1)f(X)g(X)$, where

$$f(X) = \overleftarrow{f}(X) = X^8 + X^7 + X^6 + \dots$$

$$\text{and } g(X) = \overleftarrow{g}(X).$$

- ii. Construct a $[17, 9, 5]_2$ code.
 ii. Construct a $[18, 9, 6]_2$ code.

5.8

- i. Prove that the polynomial $X^{11} + 1$ factorises in $\mathbb{F}_4[X]$ into two irreducible factors of degree 5 and one of degree 1.
 ii. Using one of the factors of degree 5, construct a $[11, 6, d]_4$ code C .
 iii. Prove that C is a $[11, 6, \geq 4]_4$ code.
 iv. With the aid of a computer, or not, verify that C is a $[11, 6, 5]_4$ code.

5.9

- i. Prove that the polynomial $X^{17} + 1$ factorises in $\mathbb{F}_4[X]$ into four irreducible factors of degree 4 and one of degree 1.
 ii. Construct a $[17, 9, \geq 7]_4$ code.
 iii. Let $g(X) = X^8 + eX^7 + X^6 + X^5 + (1 + e)X^4 + X^3 + X^2 + eX + 1$, where e is an element of \mathbb{F}_4 such that $e^2 = e + 1$. Prove that g divides $X^{17} + 1$.
 iv. Assuming that the code in ii. is $\langle g \rangle$, prove that the minimum distance of the code constructed in ii. is 7.



Maximum Distance Separable Codes

Two codewords of a block code of length n and minimum distance d must differ on any set of $n - d + 1$ coordinates, since they are at distance at least d from each other. This observation leads to the Singleton bound, [Theorem 6.1](#). A code whose parameters give an equality in the Singleton bound is called a **maximum distance separable code** or simply an **MDS code**. Therefore, an MDS code is a block code in which every possible $(n - d + 1)$ -tuple of elements of the alphabet occurs in a unique codeword for any set of $n - d + 1$ coordinates. The focus in this chapter will be on linear MDS codes, since not so much is known about non-linear MDS codes, and there are no known non-linear MDS codes which outperform linear MDS codes.

The most widely implemented linear MDS codes are the Reed–Solomon codes, whose codewords are the evaluation of polynomials of low degree. Exploiting this algebraic structure of Reed–Solomon codes will allow us to develop a fast decoding algorithm which corrects up to $\frac{1}{2}(1 - R)n$ errors, where R is the transmission rate of the code. For an arbitrary received vector, we can only correct a number of errors less than half the minimum distance. However, it may be that, although the number of errors $e \geq \frac{1}{2}d$, there is only one codeword that is at distance at most e from the received vector. We will prove that there is an algorithm, which was discovered only recently, which creates a relatively short list of possible sent codewords, when up to $(1 - \sqrt{2R})n$ errors have occurred. If we are in the afore-mentioned case that there is only one codeword close to the received vector then the list will contain only one codeword. Moreover, this list decoding algorithm can be used simultaneously for two codes which will effectively allow one to decode beyond the bound of half the minimum distance, albeit at a slightly reduced rate, as explained in [► Section 3.3](#).

The fundamental question concerning linear MDS codes asks if there are MDS codes which better the Reed–Solomon code. The MDS conjecture postulates that no such codes exist, apart from some known exceptions. The conjecture was recently proved for codes over fields of prime order but remains open over fields of non-prime order. We will prove that the longest three-dimensional linear MDS codes over a field of odd characteristic are the Reed–Solomon codes (this is not the case for even characteristic fields) and delve a little deeper into the proof to see how the tools implemented therein can be used to prove the MDS conjecture over prime fields.

6.1 Singleton Bound

Theorem 6.1 (Singleton bound)

An r -ary code C of length n and minimum distance d satisfies $|C| \leq r^{n-d+1}$.

Proof

Consider any set of $n - (d - 1)$ coordinates of a codeword. If two codewords agree on these coordinates, then their distance is at most $d - 1$. Hence, they must be different on these $n - d + 1$ coordinates. There are r^{n-d+1} distinct $(n - d + 1)$ -tuples, which gives the bound. \square

The following example is a rather trivial example of a code which meets the Singleton bound.

Example 6.2

Let A be an abelian group with r elements. Define

$$C = \{(a_1, \dots, a_{n-1}, a_1 + \dots + a_{n-1}) \mid a_i \in A\}.$$

If $a_i = b_i$ for all but one i , then $a_1 + \dots + a_{n-1} \neq b_1 + \dots + b_{n-1}$. Hence, the minimum distance of C is 2. Since $|C| = r^{n-1}$, it is an MDS code. \blacksquare

Theorem 6.3

If there is a $[n, k, d]_q$ code, then $k \leq n - d + 1$ and a $[n, k, d]_q$ code is an MDS code if and only if $k = n - d + 1$.

Proof

For a $[n, k, d]_q$ code, Theorem 6.1 implies that $q^k \leq q^{n-d+1}$. \square

6.2 Reed–Solomon Code

The Reed–Solomon code is the classical example of a linear MDS code. It is an example of an evaluation code. An **evaluation code** is a code whose codewords are the evaluation of certain functions. In the case of a Reed–Solomon code the functions are given by low degree uni-variate polynomials. In \blacktriangleright Chapter 7 and \blacktriangleright Chapter 9, we will see other examples of evaluation codes.

Example 6.4

Let $\{a_1, \dots, a_q\}$ be the set of elements of \mathbb{F}_q . The **Reed–Solomon** code is

$$C = \{(f(a_1), \dots, f(a_q), c_f) \mid f \in \mathbb{F}_q[X], \deg(f) \leq k - 1\},$$

where c_f is the coefficient of X^{k-1} in f . ■

Note that the coordinate given by c_f can be interpreted as the evaluation of f at ∞ . By homogenising the polynomial $f(X)$, we get a homogeneous polynomial

$$h(X, Y) = Y^k f(X/Y)$$

of degree k . The evaluation of f at x is $h(x, 1)$ and $h(1, 0) = c_f$.

Lemma 6.5 *The Reed–Solomon code C in Example 6.4 is a $[q + 1, k, q + 2 - k]_q$ linear MDS code.*

Proof

We will first prove that C is linear.

Let f and g be two polynomials of degree at most $k - 1$. Then $f + g$ is a polynomial of degree at most $k - 1$ and the coefficient of X^{k-1} in $f + g$ is the sum of the coefficients of X^{k-1} in f and g . Hence, if $u, v \in C$, then $u + v \in C$.

Let $\lambda \in \mathbb{F}_q$. Then λf is a polynomial of degree at most $k - 1$ and the coefficient of X^{k-1} in λf is λ times the coefficient of X^{k-1} in f . Hence, if $u \in C$, then $\lambda u \in C$.

Therefore, C is a k -dimensional linear code of length $n = q + 1$.

It follows from the fact that a non-zero polynomial of degree at most $k - 1$ has at most $k - 1$ zeros that the weight of a codeword for which $c_f \neq 0$ is at least $n - (k - 1)$. If $c_f = 0$ and $f \neq 0$, then the polynomial f has degree at most $k - 2$ and so the codeword has at most $k - 1$ zeros. Therefore, the non-zero codewords of C have weight at least $n - k + 1$. Thus, by Lemma 4.1, the code C has minimum distance at least $n - k + 1$. Then, by the Singleton bound from Theorem 6.1, C has minimum distance $n - k + 1$. Hence, C is an MDS code. □

To construct a generator matrix (g_{ij}) for the Reed–Solomon code, we choose k linearly independent polynomials $f_1(X), \dots, f_k(X)$ of degree at most $k - 1$ and index the rows with these polynomials. Then we index the columns with the elements a_1, \dots, a_q of \mathbb{F}_q . The entry $g_{ij} = f_i(a_j)$ for $j \leq q$ and the $g_{i,q+1}$ entry is the coefficient of X^{k-1} in the polynomial $f_i(X)$.

For example, with $f_i(X) = X^{i-1}$ the matrix

$$(g_{ij}) = \begin{pmatrix} 1 & 1 & \dots & 1 & 0 \\ a_1 & a_2 & \dots & a_q & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot \\ a_1^{k-2} & a_2^{k-2} & \dots & a_q^{k-2} & 0 \\ a_1^{k-1} & a_2^{k-1} & \dots & a_q^{k-1} & 1 \end{pmatrix}$$

is a generator matrix for the Reed–Solomon code.

What makes Reed–Solomon codes so attractive for implementation is the availability of fast decoding algorithms. In the following theorem, we prove that there is a decoding algorithm that will correct up to $t = \lfloor (d - 1)/2 \rfloor$ errors, where d is the minimum distance.

Although it is not really necessary, to make the proof of the following theorem easier, we shall only use the **shortened Reed–Solomon code** in which we delete the last coordinate. In this way every coordinate of a codeword is the evaluation of a polynomial at an element of \mathbb{F}_q .

Theorem 6.6

There is a decoding algorithm for a k -dimensional shortened Reed–Solomon code of length n , which corrects up to $\frac{1}{2}(n - k)$ errors and completes in a number of operations which is polynomial in n .

Proof

Suppose that we have received the vector (y_1, \dots, y_n) . We want to find the polynomial $f \in \mathbb{F}_q[X]$ of degree at most $k - 1$ such that

$$(y_1, \dots, y_n) = (f(a_1), \dots, f(a_n)) + e,$$

where e is the error vector of weight at most $\frac{1}{2}(n - k)$.

Observe that since the Reed–Solomon code is an MDS code,

$$\frac{1}{2}(n - k) = \frac{1}{2}(d - 1).$$

Let $h(X)$ be an arbitrary polynomial of degree $\lfloor \frac{1}{2}(n - k) \rfloor$ and let $g(X)$ be an arbitrary polynomial of degree $k + \lceil \frac{1}{2}(n - k) \rceil - 1$.

We determine the coefficients of g and h by solving the system of n equations,

$$g(a_j) - h(a_j)y_j = 0,$$

for $j = 1, \dots, n$. This homogeneous linear system has

$$\lfloor \frac{1}{2}(n - k) \rfloor + 1 + k + \lceil \frac{1}{2}(n - k) \rceil = n + 1$$

unknowns (the coefficients of g and h) and n equations. Hence, we can find a non-trivial solution for $h(X)$ and $g(X)$ in a number of operations that is polynomial in n using Gaussian elimination.

By assumption, there is a polynomial f of degree at most $k - 1$, such that $y_j = f(a_j)$ for at least $n - \lfloor \frac{1}{2}(n - k) \rfloor$ values of j . For these values of j , a_j is a zero of

$$g(X) - h(X)f(X).$$

The degree of this polynomial is at most $k + \lceil \frac{1}{2}(n - k) \rceil - 1$. Since

$$n - \lfloor \frac{1}{2}(n - k) \rfloor > k + \lceil \frac{1}{2}(n - k) \rceil - 1$$

the polynomial $g(X) - h(X)f(X)$ has more zeros than its degree, so it is identically zero. Therefore, $h(X)$ divides $g(X)$ and the quotient is $f(X)$. \square

In the following example we apply the algorithm in [Theorem 6.6](#) to a concrete case.

Example 6.7

Suppose that we have sent a codeword u of the 2-dimensional shortened Reed–Solomon code over $\mathbb{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$ (ordering the elements of \mathbb{F}_7 in that order) and that we have received

$$y = (1, 0, 0, 0, 6, 2, 5).$$

According to the algorithm in the proof of [Theorem 6.6](#), we should find a polynomial $g(X)$ of degree 4 and a polynomial $h(X)$ of degree 2, such that

$$g(a_j) = h(a_j)y_j,$$

for $j = 1, \dots, 7$ and where a_j is the j -th element of \mathbb{F}_7 .

The equations are

$$g(0) = h(0), \quad g(1) = 0, \quad g(2) = 0, \quad g(3) = 0, \quad g(4) = 6h(4),$$

$$g(5) = 2h(5), \quad \text{and} \quad g(6) = 5h(6).$$

From this we deduce that

$$g(X) = (X - 1)(X - 2)(X - 3)(g_1X + g_0)$$

and

$$h(X) = h_2X^2 + h_1X + h_0,$$

for some $h_2, h_1, h_0, g_1, g_0 \in \mathbb{F}_7$, which are solutions of the system

$$g_0 = h_0, \quad 6(4g_1 + g_0) = 6(2h_2 + 4h_1 + h_0)$$

$$3(5g_1 + g_0) = 2(4h_2 + 2h_1 + h_0), \quad 4(6g_1 + g_0) = 5(h_2 + 6h_1 + h_0).$$

This system of equations has a solution $g_1 = 0, g_0 = 3, h_2 = 1, h_1 = 3$ and $h_0 = 3$.

If less than $\lfloor \frac{1}{2}(n - k) \rfloor = 2$ errors have occurred, then the codeword u is the evaluation of

$$f(X) = \frac{g(X)}{h(X)} = \frac{3(X - 1)(X - 2)(X - 3)}{X^2 + 3X + 3} = 3X + 1.$$

Evaluating the polynomial f , we deduce that

$$u = (1, 4, 0, 3, 6, 2, 5).$$

■

Theorem 6.6 allows us to deduce the sent vector providing at most $\frac{1}{2}n(1 - R)$ errors occur in transmission. Recall, that the transmission rate of a k -dimensional linear code of length n is $R = k/n$. We interpreted [Exercise 3.12 v](#) as saying that (at least for a binary code) the number of codewords at a distance at most $\frac{1}{2}n(1 - \sqrt{1 - 2(d/n)})$ from a fixed vector is less than $2n$. If we consider the fixed vector as the received vector, then this implies that it should be feasible to construct a short list of possible sent codewords, even if the number of errors which have occurred exceeds half the minimum distance. A decoding algorithm which produces such a short list is called a **list decoding** algorithm. It may be that although the received codeword is more than half the minimum distance away from a codeword, it is near to only one codeword. In such a case the list decoding algorithm may allow us to correct the errors. In other words, we may be able to decode uniquely even when more than $\frac{1}{2}(d - 1)$ errors have occurred. And if we encode the message with two distinct codes, as we saw in [Section 3.3](#) this can be done so that the rate is not reduced by much, then with a high probability the intersection of the two lists will be the sent codeword.

We can list decode in a simple way by using **standard array decoding**. We make a table whose rows are indexed by the error vectors and whose columns are indexed by the codewords and whose entry is found by summing the error vector and the codeword. To decode a received vector v , one searches through the table entries for v making a list of the codewords u for which v appears in the column indexed by u . If there is a unique entry in the list, then v can be uniquely decoded. One downside to this algorithm is that for any code of reasonable size, the table requires a large amount of storage space.

The following example illustrates the main idea behind the algorithm presented in [Theorem 6.9](#), which is a list decoding algorithm for Reed–Solomon codes.

Example 6.8

Suppose that we have sent a codeword of the shortened 4-dimensional Reed–Solomon code over \mathbb{F}_{13} , where the elements of \mathbb{F}_{13} are ordered as

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}.$$

The unique decoding algorithm in [Theorem 6.6](#) allows us to correct up to $\lfloor \frac{1}{2}(d - 1) \rfloor = 4$ errors. Suppose that 5 errors have occurred and that we have received

$$y = (4, 11, 0, 3, 0, 1, 0, 0, 0, 0, 3, 12).$$

Let

$$Q(X, Y) = cY^2 + g(X)Y + h(X),$$

where $c \in \mathbb{F}_{13}$,

$$g(X) = g_4X^4 + g_3X^3 + g_2X^2 + g_1X + g_0$$

is an arbitrary polynomial of degree at most 4 and $h(X)$ is an arbitrary polynomial of degree at most 7.

We find $g(X)$ and $h(X)$, and hence $Q(X, Y)$, by solving the set of equations,

$$Q(x_j, y_j) = 0,$$

for $j = 1, \dots, 13$, where x_j is the j -th element of \mathbb{F}_{13} and y_j is the j -th coordinate of y . There are 14 unknowns in this homogeneous linear system of equations, the coefficients of g and h and the constant c , and 13 equations. Hence, there is a non-trivial solution to this system of equations.

For $x_j \in \{2, 4, 6, 7, 8, 9, 10\}$, the equation $Q(x_j, y_j) = 0$ implies $h(x_j) = 0$, since $y_j = 0$ for $j \in \{3, 5, 7, 8, 9, 10, 11\}$.

Thus,

$$h(X) = a(X - 2)(X - 4)(X - 6)(X - 7)(X - 8)(X - 9)(X - 10)$$

for some $a \in \mathbb{F}_{13}$.

The remaining equations imply,

$$4c + 11(g_0 + g_1 + g_2 + g_3 + g_4) + 10a = 0, \quad 9c + 3(g_0 + 3g_1 + 9g_2 + g_3 + 3g_4) + 11a = 0,$$

$$c + g_0 + 5g_1 + 12g_2 + 8g_3 + g_4 + 4a = 0, \quad 9c + 3(g_0 + 11g_1 + 4g_2 + 5g_3 + 3g_4) + 7a = 0,$$

and

$$c + 12(g_0 + 12g_1 + g_2 + 12g_3 + g_4) + 10a = 0.$$

Up to scalar factor, the solution of this system implies

$$\begin{aligned} Q(X, Y) &= 3Y^2 + (12X^4 + 5X^3 + 8X^2 + 7X + 5)Y \\ &+ (X - 2)(X - 4)(X - 6)(X - 7)(X - 8)(X - 9)(X - 10). \end{aligned}$$

Suppose that $f(X)$ is the polynomial of degree at most 3, whose evaluation is the sent codeword u . Then $Q(X, f(X))$ is a polynomial of degree at most 7, which is zero whenever $y_j = f(x_j)$. This occurs whenever $y_j = u_j$ and no error has occurred in the j -th coordinate. Assuming that at most 5 errors have occurred, y and u agree in at least 8 coordinates. Hence,

$Q(X, f(X))$ has at least 8 zeros. Therefore, $Q(X, f(X)) \equiv 0$ which implies $Y - f(X)$ divides $Q(X, Y)$.

Indeed, $Q(X, Y)$ factorises as

$$Q(X, Y) = (Y - X^3 - X^2 - 5X - 4)(3Y - X^4 + 8X^3 + 11X^2 + 9X + 4).$$

Since $f(X)$ is a polynomial of degree at most 3, it must be that

$$f(X) = X^3 + X^2 + 5X + 4.$$

The evaluation of $f(X)$ is

$$u = (4, 11, 0, 3, 0, 10, 2, 9, 1, 10, 3, 12),$$

which is at distance 5 from the received vector y . ■

The following theorem follows the same idea as [Example 6.8](#) and provides an algorithm which outputs a short list of possibilities for the sent codeword having received a vector in which up to approximately $n(1 - \sqrt{2R})$ errors have occurred. As with the algorithm described in [Theorem 6.6](#), this algorithm also completes in a number of operations which is polynomial in n .

Theorem 6.9

There is a decoding algorithm for a k -dimensional shortened Reed–Solomon code of length n , that completes in a number of operations which is polynomial in n and which outputs a list of less than $\sqrt{2n/(k-1)}$ codewords, one of which is the sent vector, provided less than $n - \sqrt{2nk}$ errors have occurred in transmission.

Proof

Let $m = \lceil \sqrt{2n/(k-1)} \rceil - 1$ and define a bi-variate polynomial

$$Q(X, Y) = \sum_{i=0}^m \sum_{j=0}^{\lceil \frac{1}{2}k \rceil + i(k-1)} q_{ij} X^j Y^{m-i},$$

where the coefficients are to be determined. Since

$$\begin{aligned} \sum_{i=0}^m (\lceil \frac{1}{2}k \rceil + i(k-1)) &= \lceil \frac{1}{2}k \rceil (m+1) + \frac{1}{2}m(m+1)(k-1) \\ &> \frac{1}{2}(m+1)^2(k-1) \geq n, \end{aligned}$$

the polynomial $Q(X, Y)$ has more than n coefficients.

Let (y_1, \dots, y_n) be the received vector.

The coordinates of a codeword of the shortened Reed–Solomon code are indexed by $\{x_1, \dots, x_n\}$, the set of elements of \mathbb{F}_q .

We make a homogeneous system of ℓ equations, where for each $\ell \in \{1, \dots, n\}$, we have the equation

$$Q(x_\ell, y_\ell) = 0.$$

Since we have more than n unknowns, this homogeneous system has a non-trivial solution. And we can find a solution, using Gaussian elimination, in a number of operations which is polynomial in n .

Let $g(X)$ be a polynomial of degree at most $k - 1$. Then the uni-variate polynomial $Q(X, g(X))$ has degree at most

$$\lceil \frac{1}{2}k \rceil + m(k - 1) < \sqrt{2n(k - 1)} < \sqrt{2nk}.$$

By hypothesis, less than $n - \sqrt{2nk}$ errors have occurred in transmission, so there is a polynomial $f(X)$, of degree at most $k - 1$, for which $y_\ell = f(x_\ell)$ for more than $\sqrt{2nk}$ values of ℓ . Therefore, $Q(X, f(X)) \equiv 0$, since a non-zero polynomial cannot have more zeros than its degree.

We can write

$$Q(X, Y) = (Y - f(X))C(X, Y) + R(X)$$

for some polynomials $C(X, Y)$ and $R(X)$ and conclude, substituting $Y = f(X)$, that $R(X) \equiv 0$.

Hence, $Y - f(X)$ divides $Q(X, Y)$. The bi-variate polynomial $Q(X, Y)$ can be factorised in a number of operation that is polynomial in its degree.

Thus, if $f(X)$ is the polynomial which the sent codeword is the evaluation of, then $Y - f(X)$ is a factor of $Q(X, Y)$. Since the degree in Y of $Q(X, Y)$ is at most m , there are at most m possibilities for the sent codeword.

□

6.3 Linear MDS Codes

In this section we consider the general class of linear MDS codes. For the Reed–Solomon code, the minimum distance $d = q + 2 - k$. However, for a hypothetical linear MDS code, the trivial upper bound, given by [Exercise 6.6](#), is $d \leq q$. Thus, the trivial bound does not rule out the possibility that there are linear MDS codes which are much better than Reed–Solomon codes. We will prove that linear MDS codes are equivalent to a certain geometric object and prove that, in the case that q is odd, a three-dimensional linear MDS code of length $q + 1$ is a Reed–Solomon code. This proof contains all the ingredients needed to prove that there are no linear MDS codes over fields of prime order better than the Reed–Solomon codes. The non-prime case remains open and, in part, is a more difficult problem since there are examples of linear MDS codes of length

$q + 1$ which are not equivalent to Reed–Solomon codes. These appear in [Exercise 6.9](#), [Exercise 6.10](#) and [Exercise 6.11](#).

Theorem 6.10

G is a generator matrix of a linear MDS code if and only if every subset of k columns of G is a basis of \mathbb{F}_q^k .

Proof

Let S be the multi-set of columns of the matrix G .

Suppose that G is the generator matrix of a linear MDS code. By [Lemma 4.15](#), a hyperplane of \mathbb{F}_q^k contains at most $n - d = k - 1$ vectors of S . This implies that S is a set and not a multi-set and that any k -subset of S is a basis of \mathbb{F}_q^k .

Suppose that every k -subset of S is a basis of \mathbb{F}_q^k . Then uG has at most $k - 1$ zeros for any non-zero $u \in \mathbb{F}_q^k$. Therefore, the non-zero codewords of the code C generated by G have weight at least $n - k + 1$. By [Lemma 4.1](#), the minimum weight of a non-zero codeword is equal to the minimum distance, so C has minimum distance at least $n - k + 1$. [Theorem 6.1](#) implies that the minimum distance of C is $n - k + 1$ and so C is MDS. □

Theorem 6.11

The dual of a linear MDS code is a linear MDS code.

Proof

Let C be a k -dimensional linear MDS code of length n and let $S = \{s_1, \dots, s_n\}$ be the set of columns of a generator matrix for C . The dual code C^\perp is a $(n - k)$ -dimensional linear code. Suppose $(u_1, \dots, u_n) \in C^\perp$ is a non-zero codeword of weight at most k . Since

$$u_1s_1 + \dots + u_ns_n = 0,$$

this implies that there is a linear combination of at most k of the vectors of S which are linearly dependent, contradicting [Theorem 6.10](#). Therefore, the minimum non-zero weight of C^\perp is $k + 1$. [Lemma 4.1](#) implies that C^\perp has minimum distance at least $k + 1$, which implies that C^\perp is MDS since

$$n - (n - k) + 1 = k + 1.$$

□

An **arc** is a set S of vectors of \mathbb{F}_q^k with the property that every subset of S of size k is a set of linearly independent vectors, i.e. is a basis of \mathbb{F}_q^k . Putting the vectors of an arc

of size n as the columns of a $k \times n$ matrix, one obtains a generator matrix of a linear MDS code of length n . Vice versa, the set of columns of a generator matrix of a linear MDS code is an arc of \mathbb{F}_q^k of size n . One can also consider the arc as a set of points in the projective space $\text{PG}(k-1, q)$, since the linear independence property is unchanged if we take non-zero scalar multiples of the vectors of S . Thus, an arc in $\text{PG}(k-1, q)$ is a set S of points of $\text{PG}(k-1, q)$ with the property that every subset of size k spans the whole space.

Theorem 6.12

If $k \geq q$, then a k -dimensional linear MDS code over \mathbb{F}_q has minimum distance at most 2.

Proof

Suppose that the minimum distance of a linear MDS code C of length n is at least 3. By Theorem 6.3, $n = d + k - 1 \geq k + 2$.

By Theorem 6.10, the set S of columns of a generator matrix of C is an arc of \mathbb{F}_q^k . Order the vectors in S arbitrarily and let

$$B' = \{e_1, \dots, e_k\}$$

be the first k vectors of S .

Consider the coordinates of the $(k+1)$ -st and the $(k+2)$ -nd vector of S with respect to the basis B' . Suppose there is a zero in the i -th coordinate of one of them. Then the hyperplane, defined by $X_i = 0$, contains k vectors of S (since it contains $k-1$ vectors of B'), contradicting the arc property.

Therefore, we can find λ_j , non-zero elements of \mathbb{F}_q , such that the $(k+1)$ -st vector in S is the all-one vector with respect to the basis

$$B = \{\lambda_1 e_1, \dots, \lambda_k e_k\}.$$

Let (s_1, \dots, s_k) be coordinates of the $(k+2)$ -nd vector of S with respect to the basis B . As observed above $s_i \neq 0$, for all $i = 1, \dots, k$. Since $k > q - 1$, the pigeon-hole principle implies there exists an i and j such that $s_i = s_j$. Therefore, the hyperplane defined by the equation $X_i = X_j$ contains k vectors of S , again contradicting the arc property. \square

We can obtain a binary code from a linear code over \mathbb{F}_q by identifying each element of \mathbb{F}_q with a binary string of length $\lceil \log_2 q \rceil$. In this way, if we take a k -dimensional linear code of length N over \mathbb{F}_q , then we get a binary code of length $n = N \lceil \log_2 q \rceil$ with q^k elements. The rate of the binary code is $\log_2 |C|/n \approx k/N$ and the relative minimum distance is approximately $d/(N \log_2 q)$.

From a linear MDS code we obtain a binary code with a transmission rate R of approximately k/N , whose relative minimum distance is approximately

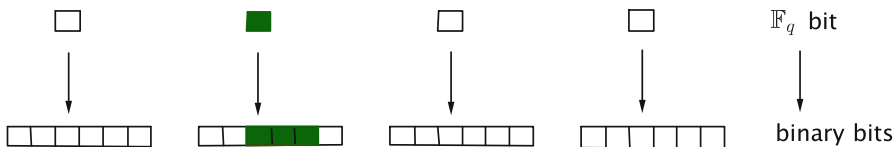


Fig. 6.1 A burst of three errors distorting only one \mathbb{F}_q bit.

$$\frac{1 - R}{\log_2 q} + \frac{1}{n}.$$

If $k \geq q$, then according to Theorem 6.12 the minimum distance is at most 2, so we assume $k < q$. But then Exercise 6.6 implies $N < 2q - 1$ which implies that $n < 2q \log_2 q$. If we want n to tend to infinity, we must have q going to infinity, which implies that if the rate is bounded away from zero, then the relative minimum distance will tend to zero. Hence, we cannot hope to obtain asymptotically good codes in this way. This does not mean that MDS codes and in particular Reed–Solomon codes are not used in practice. Although as a binary code, a Reed–Solomon code can only guarantee the correction of a relatively small amount of errors, in certain circumstances it can correct many more errors.

Consider Figure 6.1. Each element of \mathbb{F}_q is mapped to a string of $\lceil \log_2 q \rceil$ binary bits. If all the errors in the transmission of the binary string occur in the same run of bits, then when we look to decode the \mathbb{F}_q -linear code, only one bit has been distorted. This is particularly useful for channels in which errors tend to come in bursts. For this reason Reed–Solomon codes, and more generally linear codes over large fields, are very useful for **burst-error** correction. Furthermore, a common method in the application of Reed–Solomon codes is to simultaneously use two codes of high rate. For example, in a CD the data is encoded using two Reed–Solomon codes over \mathbb{F}_{64} which are cross-interleaved. Add to that the fast decoding algorithms which allow us to decode past the half the minimum distance bound with high probability and one has a fast and efficient means of error-correction.

6.4 MDS Conjecture

Theorem 6.12 implies that we should only look for linear MDS codes of dimension $k \leq q - 1$. Theorem 6.11 implies that a k -dimensional MDS code of length n exists if and only if a $(n - k)$ -dimensional MDS code of length n exists. By Exercise 6.6, a 2-dimensional MDS code has length at most $q + 1$ and a 3-dimensional MDS code has length at most $q + 2$. By Theorem 6.11, the dual of a $(q - 1)$ -dimensional MDS code of length $q + 2$ is a 3-dimensional MDS code of length $q + 2$, which by Exercise 6.7 does not exist for q odd and, by Exercise 6.9, does exist for q even. From this we can deduce the length of the longest linear MDS codes for all dimensions not in the range

$4 \leq k \leq q - 2$. The **MDS conjecture** asserts that, within this range, one cannot do better than the Reed–Solomon code.

Conjecture 6.13 (MDS conjecture) If $4 \leq k \leq q - 2$, then a k -dimensional linear MDS code of length n satisfies $n \leq q + 1$.

The MDS conjecture has been verified for q prime. It follows from the following theorem, whose proof we will sketch at the end of this section.

Theorem 6.14

Let $q = p^h$, where p is prime. If $k \leq p$, then a k -dimensional linear MDS code of length n satisfies $n \leq q + 1$.

We will prove the following theorem only in the case $k = 3$. In this case the hypothesis $k \neq \frac{1}{2}(q + 1)$ is not necessary. The hypothesis $k \leq p$ is necessary. The following example is not equivalent to a Reed–Solomon code.

Example 6.15

Let $\mathbb{F}_{32} = \{a_1, \dots, a_{32}\}$ and let C be the three-dimensional linear code over \mathbb{F}_{32} generated by the matrix G whose i -th column is $(1, a_i, a_i^4)^t$, for $i = 1, \dots, 32$, and whose 33-rd column is $(0, 0, 1)^t$. A generic 3×3 submatrix of G is of the form

$$\begin{pmatrix} 1 & 1 & 1 \\ x & y & z \\ x^4 & y^4 & z^4 \end{pmatrix}.$$

The determinant of this matrix is

$$(z - x)^4(y - x) - (y - x)^4(z - x)$$

If this determinant is zero, then $((z - x)/(y - x))^3 = 1$. Since the field \mathbb{F}_{32} contains no element $e \neq 1$ such that $e^3 = 1$, this implies that $z = y$. In this way we see that all 3×3 submatrices of G are non-singular and, by [Theorem 6.10](#), C is an MDS code. ■

Theorem 6.16

Let $q = p^h$, where p is prime. If $k \leq p$ and $k \neq \frac{1}{2}(q+1)$, then a k -dimensional linear MDS code of length $q+1$ is a Reed–Solomon code.

Proof (for $k = 3$)

Let S be the set of columns of a generator matrix G of a 3-dimensional linear MDS code C of length $q+1$. By [Theorem 6.10](#), a hyperplane of \mathbb{F}_q^3 contains at most two vectors of S .

Let $s \in S$. There are $q+1$ hyperplanes containing s , q of which contain a vector of $S \setminus \{s\}$. Hence, there is exactly one hyperplane of \mathbb{F}_q^3 which contains s and no other vector of S .

Let $f_s(X)$ be a linear form whose kernel is this hyperplane.

Let $x, y, z \in S$. We claim that

$$f_x(y)f_y(z)f_z(x) = f_y(x)f_z(y)f_x(z).$$

With respect to the basis $B = \{x, y, z\}$, the hyperplane that contains $s = (s_1, s_2, s_3)$ and $x = (1, 0, 0)$ is the kernel of the linear form

$$X_2 - (s_2/s_3)X_3.$$

Note that $s_3 \neq 0$ since the hyperplane $\ker X_3$ is incident with x and y .

The linear form

$$f_x(X) = a_2X_2 + a_3X_3,$$

for some $a_2, a_3 \in \mathbb{F}_q$, since the kernel of $f_x(X)$ contains x .

For distinct $s \in S \setminus \{x, y, z\}$, the value of s_2/s_3 is distinct, since the linear form

$$X_2 - (s_2/s_3)X_3$$

is different, for different $s \in S \setminus \{x, y, z\}$.

Since,

$$X_2 - (s_2/s_3)X_3 \neq X_2 + (a_3/a_2)X_3,$$

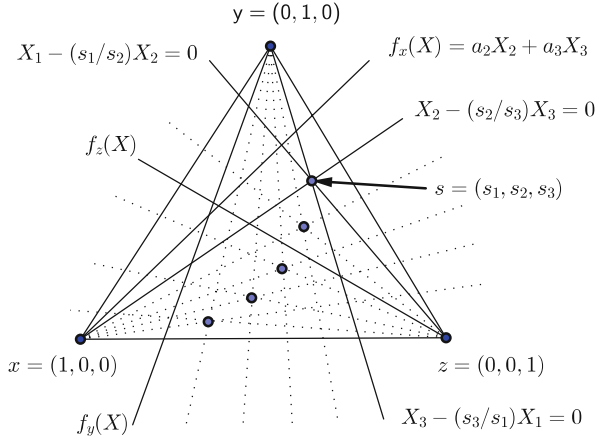
for any $s \in S \setminus \{x, y, z\}$, we have that

$$\left\{ \frac{s_2}{s_3} \mid s \in S \setminus \{x, y, z\} \right\} \cup \left\{ -\frac{a_3}{a_2} \right\}$$

is the set of all non-zero elements of \mathbb{F}_q .

In [Figure 6.2](#), the vectors x, y and z are drawn as points in $\text{PG}(2, q)$ and the hyperplanes defined as the kernels of f_x, f_y and f_z are lines in the plane.

Note that $a_2 = f_x(y)$ and $a_3 = f_x(z)$. The product of all the non-zero elements of \mathbb{F}_q is -1 , so we have that



■ **Fig. 6.2** The lines joining the basis points to s and the tangent lines.

$$-\frac{f_x(z)}{f_x(y)} \prod_{s \in S \setminus B} \frac{s_2}{s_3} = -1.$$

Similarly,

$$\frac{f_y(x)}{f_y(z)} \prod_{s \in S \setminus B} \frac{s_3}{s_1} = 1 \quad \text{and} \quad \frac{f_z(y)}{f_z(x)} \prod_{s \in S \setminus B} \frac{s_1}{s_2} = 1.$$

Multiplying the three equations together establishes the claim.

For any $s \in S$, the linear form $f_s(X)$ is determined by its value at the basis elements, so

$$f_s(X) = f_s(x)X_1 + f_s(y)X_2 + f_s(z)X_3.$$

Evaluating at $X = s$, and using the fact that $f_s(s) = 0$,

$$f_s(x)s_1 + f_s(y)s_2 + f_s(z)s_3 = 0.$$

By the claim,

$$s_1 + \frac{f_y(s)f_x(y)}{f_x(s)f_y(x)}s_2 + \frac{f_z(s)f_x(z)}{f_x(s)f_z(x)}s_3 = 0.$$

Now, substituting

$$f_x(s) = f_x(y)s_2 + f_x(z)s_3,$$

$$f_y(s) = f_y(x)s_1 + f_y(z)s_3,$$

$$f_z(s) = f_z(x)s_1 + f_z(y)s_2,$$

we get

$$2(c_3s_1s_2 + c_2s_1s_3 + c_1s_2s_3) = 0,$$

for some $c_1, c_2, c_3 \in \mathbb{F}_q \setminus \{0\}$.

Explicitly,

$$c_1 = f_y(z) \frac{f_x(y)}{f_y(x)}, \quad c_2 = f_x(z), \quad \text{and} \quad c_3 = f_x(y).$$

Since we are assuming that q is odd, the vectors of S are zeros of the quadratic form

$$c_3X_1X_2 + c_2X_1X_3 + c_1X_2X_3.$$

The zeros of this quadratic form excluding $(0, 0, 1)$ are parameterisable by

$$s = \left(t, \frac{c_2}{c_1^2}(c_3 - c_1t), \frac{t}{c_1}(c_1t - c_3) \right).$$

Therefore, the i -th coordinate of a column of G is the evaluation of the polynomial $f_i(X)$, where

$$f_1(X) = X, \quad f_2(X) = -c_2c_1^{-1}X + c_1^{-2}c_2c_3$$

and

$$f_3(X) = X^2 - c_1^{-1}c_3X,$$

with the exception of the column $(0, 0, 1)$, whose i -th coordinate is the coefficient of X^2 of $f_i(X)$.

Hence, the codeword $(u_1, u_2, u_3)G$ is the evaluation of the polynomial

$$u_1f_1(X) + u_2f_2(X) + u_3f_3(X),$$

so C is a Reed–Solomon code. □

The proof of [Theorem 6.14](#) and the general proof of [Theorem 6.16](#) follow the same strategy as the proof of [Theorem 6.16](#) given here for $k = 3$.

Let S be the set of columns of a generator matrix of a k -dimensional linear MDS code over \mathbb{F}_q . Let $\tau = q + k - 1 - |S|$.

Let A be a $(k - 2)$ -subset of S . There are $q + 1$ hyperplanes which contain the $(k - 2)$ -dimensional subspace of \mathbb{F}_q^k spanned by the points of A , τ of which contain no other vectors of S . With linear forms $\alpha_1, \dots, \alpha_\tau$, whose kernels are these τ hyperplanes, we define a polynomial

$$f_A(X) = \prod_{i=1}^{\tau} \alpha_i(X).$$

This defines $f_A(X)$ uniquely up to scalar factor. The claim is then that

$$f_{D \cup \{x\}}(y) f_{D \cup \{y\}}(z) f_{D \cup \{z\}}(x) = (-1)^{\tau+1} f_{D \cup \{y\}}(x) f_{D \cup \{z\}}(y) f_{D \cup \{x\}}(z),$$

for all k -subsets $D \cup \{x, y, z\}$ of S .

To simplify matters slightly, let us assume that τ is odd. Since the polynomials $f_A(X)$ are defined up to scalar factor, we can scale them in such a way that the above equality gives

$$f_{D \cup \{x\}}(y) = f_{D \cup \{y\}}(x).$$

This implies that for C , a $(k-1)$ -subset of S , there is a non-zero $a_C \in \mathbb{F}_q$, such that for all $e \in C$,

$$f_{C \setminus \{e\}}(e) = a_C.$$

The interpolation of the linear form $f_s(X)$ in the proof for $k=3$ is generalised to interpolating the polynomial $f_A(X)$ of degree τ . Even though $f_A(X)$ is a homogeneous polynomial in k variables, since it is the product of linear forms whose kernels contain the subspace spanned by $A = \{a_1, \dots, a_{k-2}\}$, with respect to a basis of \mathbb{F}_q^k containing A , $f_A(X)$ is a homogeneous polynomial in two variables. Hence, it can be interpolated. To be able to interpolate we fix a subset E of S of size $k + \tau$ and an element $x \in E$. Writing $\det(X, u, A)$ as a shorthand for

$$\begin{vmatrix} X_1 & X_2 & \dots & X_k \\ u_1 & u_2 & \dots & u_k \\ a_{11} & a_{12} & \dots & a_{1k} \\ \vdots & \vdots & \dots & \vdots \\ a_{k-2,1} & a_{k-2,2} & \dots & a_{k-2,k} \end{vmatrix}$$

the interpolation implies

$$f_A(X) = \sum_{e \in E \setminus (A \cup \{x\})} f_A(e) \prod_{u \in E \setminus (A \cup \{x, e\})} \frac{\det(X, u, A)}{\det(e, u, A)}.$$

One can check that substituting $X = e$ gives $f_A(e)$ on the right-hand side for all $e \in E \setminus \{x\}$.

Substituting $X = x$ and rearranging terms leads to the equation

$$\sum_{e \in E \setminus A} f_A(e) \prod_{u \in E \setminus (A \cup \{e\})} \det(u, e, A)^{-1} = 0.$$

This implies that for a $(k - 2)$ -subset A of E ,

$$\sum_{C \supset A} a_C \prod_{u \in E \setminus C} \det(u, C)^{-1} = 0,$$

where the sum runs over the $(k - 1)$ -subsets C of E containing A .

Thus, we get an equation for each $(k - 2)$ -subset A of E . Setting

$$\lambda_C = a_C \prod_{u \in E \setminus C} \det(u, C)^{-1},$$

the equation is simply

$$\sum_{C \supset A} \lambda_C = 0,$$

where the sum runs over the $(k - 1)$ -subsets C of E containing A .

This set of equations is enough to prove both [Theorem 6.14](#) and, with a little more work, [Theorem 6.16](#). To prove [Theorem 6.14](#), we assume that the length of the MDS code is $q + 2$ and so $|S| = q + 2$. We can assume by [Theorem 6.11](#) that $k \leq (q + 2)/2$, taking the dual code if necessary. Since $|S| = q + 2$ we have that $\tau = k - 3$ and $|E| = 2k - 3$. There are $N = \binom{2k-3}{k-2}$ linear equations. For each $(k - 1)$ -subset C of E , we have an unknown λ_C , so in all we have $\binom{2k-3}{k-1}$ unknowns. Thus, we have a linear system of N equations in N unknowns. This system of equations implies the equation

$$(k - 1)! \lambda_C = 0,$$

which is a contradiction if $k \leq p$, since $\lambda_C \neq 0$.

6.5 Comments

The Singleton bound appears in Singleton's paper [66] on MDS codes from 1964, the Reed–Solomon codes having already been published some years before in [60]. In the same paper the authors detail the decoding algorithm presented in [Theorem 6.6](#). An algorithm based on interpolation was proposed by Berlekamp and Welch [9]. As mentioned in the text, the list decoding of [Theorem 6.9](#), from Sudan [68], decodes up to $(1 - \sqrt{2R})n$ errors and produces a list with a constant number of codewords. This bound improves on the unique decoding bound of $\frac{1}{2}(1 - R)n$ when $R < 3 - 2\sqrt{2}$. The bound can be improved to $(1 - \sqrt{R})n$ by interpolating the zeros with multiplicity, see Guruswami and Sudan [33]. The bound $(1 - \sqrt{R})n$ is larger than the unique decoding bound for all R . There is a limit of $(1 - R)n$ errors, beyond which one cannot hope to produce a list with a constant number of codewords, see Guruswami and Rudra [32]. Indeed, one can find received vectors for which the number of codewords of the Reed–Solomon code, at a distance of $(1 - R)n$ or less, is not bounded by a polynomial in n .

One can use a Reed–Solomon code, or more generally an MDS code, in concatenation codes to produce binary codes which come arbitrarily close to the Gilbert–Varshamov bound asymptotically. This construction is due to Thommesen [70]. Suppose we have an MDS code of length N and rate R over a field with 2^k elements. A codeword is a vector $v = (v_1, \dots, v_N) \in \mathbb{F}_{2^k}^N$. For each coordinate, we randomly choose G_i , a $k \times n$ matrix with entries from \mathbb{F}_2 . We consider each coordinate v_i of v as a vector in \mathbb{F}_2^k . Recall that, in ► Chapter 2, we constructed the elements of \mathbb{F}_{2^k} as elements of $\mathbb{F}_2[X]/(f)$, where $f(X)$ is an irreducible polynomial of degree k in $\mathbb{F}_2[X]$. If

$$v_i = a_0 + a_1X + \dots + a_{k-1}X^{k-1}$$

in this quotient ring, then we consider v_i as the vector $(a_0, a_1, \dots, a_{k-1})$. We then make a binary code of length nN by taking as its elements the strings

$$u = (v_1G_1, v_2G_2, \dots, v_NG_N).$$

With a high probability this binary code will be arbitrarily close to the Gilbert–Varshamov bound.

For more on applications of Reed–Solomon codes, in particular the simultaneous use of two Reed–Solomon codes in compact discs, see [76].

MDS codes are widely used in distributed storage systems to protect data from server failures. A codeword vG of a k -dimensional linear MDS code can be recovered from just k coordinates, since k of the coordinates uniquely determine v and therefore the codeword. This recoverability property is the important feature of local reconstruction codes, of which MDS codes are a special sub-class. A **local reconstruction code** is a block code in which for each coordinate i , there is a subset of the coordinates R_i , such that knowing the coordinates in R_i of a codeword, one can recover the i -th coordinate of the codeword. It is this more general class of codes which are implemented in distributed storage systems.

The MDS conjecture appears in [50] although its origins can be traced back to the fundamental questions asked by Segre [64] in 1967. Proofs of Theorem 6.14 and Theorem 6.16 can be found in Ball [5], although the original proof of Theorem 6.16 for $k = 3$ is due to Segre [63].

The MDS code in Exercise 6.10 was discovered by Glynn [26], and the MDS code in Exercise 6.11 is due to Segre [64].

6.6 Exercises

6.1 Prove that the dual of a Reed–Solomon code is a Reed–Solomon code.

6.2 Shortening the k -dimensional Reed–Solomon code over \mathbb{F}_q , by removing the last column and the column which is the evaluation of the polynomial at zero in Example 6.4, we get a k -dimensional linear code C of length $q - 1$. Prove that C is the cyclic code in Example 5.12.

6.3 Let C be the 4-dimensional shortened Reed–Solomon code of length 7 over \mathbb{F}_7 which is the evaluation of polynomials of degree at most three, where the elements of \mathbb{F}_7 are ordered as $\{0, 1, 2, 3, 4, 5, 6\}$. Decode the received vector $(1, 1, 0, 4, 2, 2, 1)$ using the algorithm from [Theorem 6.6](#).

6.4 Suppose that E is the set of coordinates where an error has occurred in transmission. Prove that in the set of equations in [Theorem 6.6](#), $h(a_i) = 0$ for all $i \in E$.

6.5 Let C be the linear code from [Exercise 4.8](#). Prove that C^\perp is an MDS code and that therefore C is an MDS code.

6.6 Prove that if there exists an $[n, k, n - k + 1]_q$ code, then $n \leq q + k - 1$.

6.7 Let S be the set of columns of a generator matrix of a 3-dimensional linear MDS code of length n , considered as a set of points of $\text{PG}(2, q)$.

- i. Prove that S is a set of n points, no three of which are collinear.
- ii. Prove that if q is odd then a 3-dimensional linear MDS code has length at most $q + 1$.
- iii. Prove that if q is even, then a 3-dimensional linear MDS code of length $q + 1$ is extendable to a 3-dimensional linear MDS code of length $q + 2$.

6.8 Let S be the set of columns of a generator matrix of a 4-dimensional linear MDS code of length n , considered as a set of points of $\text{PG}(3, q)$.

- i. Prove that S is a set of n points, no 4 of which are contained in a hyperplane.
- ii. Prove that if $|S| = q + 2$ and q is even, then to each point x of S there is a line ℓ_x , incident with x , with the property that each plane containing ℓ_x contains exactly one point of $S \setminus \{x\}$.
- iii. Prove that there are no 4-dimensional linear MDS codes of length $q + 3$.

6.9 Show that if e and h are co-prime, then the matrix whose columns are $\{(1, t, t^{2^e}) \mid t \in \mathbb{F}_q\} \cup \{(0, 1, 0), (0, 0, 1)\}$ generates a 3-dimensional linear MDS of length $q + 2$ over \mathbb{F}_q , $q = 2^h$.

6.10 Show that the matrix whose columns are $\{(1, t, t^2 + \eta t^6, t^3, t^4) \mid t \in \mathbb{F}_9\} \cup \{(0, 0, 0, 0, 1)\}$, where $\eta^4 = -1$, generates a 5-dimensional MDS of length 10 over \mathbb{F}_9 .

6.11 Suppose q is even and let σ be an automorphism of \mathbb{F}_q . Let

$$M = \begin{pmatrix} e^{\sigma+1} & e^\sigma b & eb^\sigma & b^{\sigma+1} \\ e^\sigma c & e^\sigma d & b^\sigma c & b^\sigma d \\ c^\sigma e & c^\sigma b & d^\sigma e & d^\sigma b \\ e^{\sigma+1} & c^\sigma d & cd^\sigma & d^{\sigma+1} \end{pmatrix}.$$

Let A be the 4×4 matrix whose i -th column is the transpose of $(1, t_i, t_i^\sigma, t_i^{\sigma+1})$.

- i. Verify that if e, b, c, d are chosen so that $c + dt_1 = 0$, $e + bt_2 = 0$ and $e + bt_3 = c + dt_3$, then

$$\det \mathbf{MA} = ((e + bt_1)(c + dt_2)(e + bt_3))^{\sigma+1} \det \begin{pmatrix} 1 & 0 & 1 & (e + bt_4)^{\sigma+1} \\ 0 & 0 & 1 & (e + bt_4)^{\sigma} (c + dt_4) \\ 0 & 0 & 1 & (c + dt_4)^{\sigma} (e + bt_4) \\ 0 & 1 & 1 & (c + dt_4)^{\sigma+1} \end{pmatrix}.$$

- ii. Prove that if there is no non-zero element $a \in \mathbb{F}_q$ for which $a^{\sigma} = a$, then the code generated by the matrix \mathbf{G} , whose columns are the transpose of $(1, t, t^{\sigma}, t^{\sigma+1})$, for each $t \in \mathbb{F}_q$, is a 4-dimensional linear MDS code of length q .
- iii. Prove that by adding the column $(0, 0, 0, 1)^t$ to \mathbf{G} , the code generated by the matrix extends the 4-dimensional linear MDS code of length q to a 4-dimensional linear MDS code of length $q + 1$.



Alternant and Algebraic Geometric Codes

Alternant codes are subfield subcodes of a generalised Reed–Solomon code over an extension field of \mathbb{F}_q . This is a large class of linear codes which includes BCH codes, one of the families of cyclic codes which appeared in ► Chapter 5. Although BCH codes are not asymptotically good, we will prove that there are asymptotically good alternant codes. Not only are alternant codes linear, and so easy to encode, they also have an algebraic structure which can be exploited in decoding algorithms. However, as with the codes constructed in Theorem 3.7, the construction of these asymptotically good alternant codes is probabilistic. We prove that such a code must exist without giving an explicit construction.

Algebraic geometric codes are codes constructed from algebraic curves. We shall cover the basic properties of these codes and prove that algebraic geometric codes can provide examples of asymptotically good codes. In the case of r -ary codes, where r is the square of an odd prime larger than or equal to 7, there are algebraic geometric codes whose rate and relative minimum distance exceed the Gilbert–Varshamov bound.

7.1 Subfield Subcodes

Let C be a linear code over \mathbb{F}_{q^h} of length n .

The **subfield subcode** $A(C)$ of C is a code over \mathbb{F}_q , defined as the set of codewords of C all of whose coordinates are elements of \mathbb{F}_q .

Lemma 7.1 *If C is a $[n, k', d]_{q^h}$ code, then $A(C)$ is a $[n, k, \geq d]_q$ code, where $k' \geq k \geq n - (n - k')h$.*

Proof

Suppose $u, v \in A(C)$. Then $u, v \in C$ and since C is linear $u + v \in C$. Since $u + v$ has coordinates in \mathbb{F}_q , $u + v \in A(C)$. Similarly $\lambda u \in A(C)$, for all $\lambda \in \mathbb{F}_q$, so $A(C)$ is linear over \mathbb{F}_q .

Let H be a $(n - k') \times n$ check matrix for C . Then $(a_1, \dots, a_n) \in A(C)$ if and only if

$$a_1x_1 + \dots + a_nx_n = 0,$$

for all rows (x_1, \dots, x_n) of H .

As we saw in ► Chapter 2, the elements x_i of \mathbb{F}_q^h are polynomials in the ring $\mathbb{F}_q[X]/(f)$, where f is an irreducible polynomial of degree h in $\mathbb{F}_q[X]$. Writing the equation above over $\mathbb{F}_q[X]$ gives at most h constraints ($j = 1, \dots, h$) on $A(C)$ of the form

$$a_1x_{1j} + \dots + a_nx_{nj} = 0,$$

where $x_{ij} \in \mathbb{F}_q$ is defined by

$$x_i = x_{i1} + x_{i2}X + \dots + x_{ih}X^{h-1}.$$

Therefore, a check matrix for $A(C)$ has rank at most $(n - k')h$, which implies that the dimension of $A(C)$ is at least $n - (n - k')h$. □

Example 7.2

Let e be a primitive element of \mathbb{F}_9 which satisfies $e^2 = e + 1$.

Consider the check matrix

$$H = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 & 0 \\ e & e^2 & \dots & e^8 & 0 & 1 \end{pmatrix}$$

of a $[10, 8, 3]_9$ code C .

Writing out the elements of H as $a + be$, where $a, b \in \mathbb{F}_3$, we get a check matrix

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

for $A(C)$ over \mathbb{F}_3 . The rank of this matrix is 3, so $A(C)$ has dimension 7. A non-zero codeword of C has weight at least 3, so it follows that a non-zero codeword of $A(C)$ has weight at least 3. Thus, $A(C)$ is a $[10, 7, \geq 3]_3$ code and since $(0, 0, 0, 0, 0, 0, 1, 2, 2) \in A(C)$, $A(C)$ is a $[10, 7, 3]_3$ code. ■

7.2 Generalised Reed–Solomon Codes

Let x_1, \dots, x_n be distinct elements of \mathbb{F}_{q^h} and let v_1, \dots, v_n be non-zero elements of \mathbb{F}_{q^h} .

The linear code over \mathbb{F}_{q^h} defined by

$$C = \{(v_1 f(x_1), \dots, v_n f(x_n)) \mid f \in \mathbb{F}_{q^h}[X], \deg f \leq k' - 1\},$$

where $k' < q^h$, is a **generalised Reed–Solomon** (GRS) code.

The subfield subcode $A(C)$ is an **alternant code** if C is a GRS code. Alternant codes are useful because they allow us to construct linear codes over small fields (including binary codes). Not only can they have good parameters, the algebraic structure, inherent in their construction, can be exploited to develop reasonably fast decoding algorithms.

Example 7.3

Let e be a primitive element of \mathbb{F}_8 , such that $e^3 = e + 1$ and let C be the GRS code

$$\{(f(0), f(1), f(e), e^6 f(e^2), f(e^3), e^4 f(e^4), e^5 f(e^5), e^3 f(e^6)) \mid f \in \mathbb{F}_8[X], \deg f \leq 5\}.$$

The matrix

$$\begin{pmatrix} 1 & 1 & 1 & e & 1 & e^3 & e^2 & e^4 \\ 0 & 1 & e & e^3 & e^3 & 1 & 1 & e^3 \end{pmatrix}$$

is a check matrix for C . This can readily be checked. The scalar product of the first and second rows of the matrix with a codeword of C is

$$\sum_{x \in \mathbb{F}_8} f(x) \quad \text{and} \quad \sum_{x \in \mathbb{F}_8} x f(x)$$

respectively. For a polynomial f of degree at most 5 these sums are zero, see [Exercise 2.8](#).

As in [Example 7.2](#), we write out the elements of H as $a + be + ce^2$, where $a, b, c \in \mathbb{F}_2$. In this way, we get a check matrix

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

for $A(C)$. The matrix H has rank 5, so the dimension of $A(C)$ is 3.

By solving the system of equations $Hu^t = 0$, where $u \in \mathbb{F}_2^8$, we can find a basis for the code $A(C)$ and therefore a generator matrix G for $A(C)$. One can readily check that $GH^t = 0$, where

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

The seven non-zero codewords of $A(C)$ have weight at least 3. By [Lemma 4.1](#), the minimum distance of $A(C)$ is 3, since this is minimum weight of a non-zero codeword of $A(C)$. Therefore, $A(C)$ is a $[8, 3, 3]_2$ code. ■

Lemma 7.4 *If C is a k' -dimensional GRS code of length n over \mathbb{F}_{q^h} , then the minimum distance of $A(C)$ is at least $n - k' + 1$.*

Proof

As for Reed–Solomon codes, since a non-zero polynomial of degree at most $k' - 1$ has at most $k' - 1$ zeros, a non-zero codeword of the code C has weight at least $n - (k' - 1)$. Therefore, the weight of any non-zero codeword of $A(C)$ is at least $n - k' + 1$. By [Lemma 4.1](#), the minimum weight of a non-zero codeword is equal to the minimum distance. □

The following lemma is Lagrange interpolation, which will be used to prove [Lemma 7.6](#).

Lemma 7.5 *Suppose that a_1, \dots, a_k are distinct elements of \mathbb{F}_q and let b_1, \dots, b_k be elements of \mathbb{F}_q . There is a unique polynomial $f \in \mathbb{F}_q[X]$ of degree at most $k - 1$ such that $f(a_i) = b_i$, for all $i = 1, \dots, k$.*

Proof

For each $i = 1, \dots, k$, the equality $f(a_i) = b_i$ is a constraint on the polynomial $f(X) = \sum_{i=1}^{k-1} c_i X^i$,

$$c_0 + c_1 a_i + \dots + c_{k-1} a_i^{k-1} = b_i.$$

In matrix form this system of equations is

$$\begin{pmatrix} 1 & a_1 & \dots & a_1^{k-1} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 1 & a_k & \dots & a_k^{k-1} \end{pmatrix} \begin{pmatrix} c_0 \\ \cdot \\ \cdot \\ \cdot \\ c_{k-1} \end{pmatrix} = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_k \end{pmatrix}.$$

Since the matrix is a Vandermonde matrix and the a_i 's are distinct, the matrix has non-zero determinant and so the system of equations has a unique solution. □

In [Theorem 7.7](#), we are going to prove that there are alternant codes which do not have any non-zero codewords of small weight. We do this by counting, bounding by above the number of alternant codes containing vectors of small weight d . Then, by proving there are more alternant codes than alternant codes containing vectors of weight less than d , we conclude that there are alternant codes with minimum distance at least

d. Firstly, we bound the number of alternant codes containing a fixed non-zero vector of \mathbb{F}_q^n .

Lemma 7.6 *Let a be a non-zero vector of \mathbb{F}_q^n . The number of k' -dimensional GRS codes of length n over \mathbb{F}_{q^h} containing a , for a fixed n -tuple (x_1, \dots, x_n) of distinct elements of \mathbb{F}_{q^h} , is at most $(q^h - 1)^{k'}$.*

Proof

If there is a k' -dimensional GRS codes of length n over \mathbb{F}_{q^h} containing a , then a has at most $k' - 1$ zero coordinates. After a suitable permutation of the coordinates, assume that all the zero coordinates of a are contained in the first $k' - 1$ coordinates. Choose $v_1, \dots, v_{k'} \in \mathbb{F}_{q^h} \setminus \{0\}$.

So that a is contained in the GRS code with this choice of $v_1, \dots, v_{k'}$, there must be a polynomial f of degree at most $k' - 1$ such that

$$f(x_i) = \frac{a_i}{v_i},$$

for $i = 1, \dots, k'$. By Lemma 7.5, there is a unique polynomial f with this property.

For $j = k' + 1, \dots, n$, if $f(x_j) = 0$, then a is not contained in a GRS code for this choice of $v_1, \dots, v_{k'}$, since the zeros of a all occur in the first $k' - 1$ coordinates. Hence, $f(x_j) \neq 0$ and the elements v_j are fixed by

$$v_j = \frac{a_j}{f(x_j)}.$$

Moreover, $v_j \neq 0$ since $a_j \neq 0$ for these values of j . □

7.3 Alternant Codes Meeting the Gilbert–Varshamov Bound

We now prove that there are alternant codes of rate R and minimum distance d whose parameters approximate to the parameters of a code on the Gilbert–Varshamov curve of Figure 3.1.

Theorem 7.7

There are asymptotically good alternant codes of rate $R \in \mathbb{Q}$ meeting the Gilbert–Varshamov bound.

Proof

Choose h and n so that Rn is an integer, h divides $n - Rn$ and

$$k' = n - \frac{n(1 - R)}{h} < q^h.$$

By [Lemma 7.6](#), for a fixed x_1, \dots, x_n , the number of k' -dimensional GRS codes over \mathbb{F}_{q^h} of length n containing the vectors of \mathbb{F}_q^n of weight at most $d - 1$ is at most

$$(q^h - 1)^{k'} \sum_{j=0}^{d-1} (q - 1)^j \binom{n}{j}.$$

Since we can choose $v_1, \dots, v_n \in \mathbb{F}_{q^h} \setminus \{0\}$, the total number of k' -dimensional GRS codes for a fixed x_1, \dots, x_n is $(q^h - 1)^n$.

Therefore, if

$$(q^h - 1)^{k'} \sum_{j=0}^{d-1} (q - 1)^j \binom{n}{j} < (q^h - 1)^n,$$

then there is a k' -dimensional GRS code C with no non-zero codewords of weight less than d .

By [Lemma 7.1](#), the alternant code $A(C)$ has dimension $k \geq n - (n - k')h = Rn$, so the rate of $A(C)$ is at least R . Substituting $k' = n - n(1 - R)/h$, the condition is that

$$(q^h - 1)^{Rn/h} \sum_{j=0}^{d-1} (q - 1)^j \binom{n}{j} < (q^h - 1)^{n/h},$$

which is (asymptotically) the Gilbert–Varshamov bound, since $(q^h - 1)^{r/h} \approx q^r$. \square

The following theorem says that we can decode received vectors with few errors quickly. Observe that the minimum distance d , for the alternant code which we get from [Theorem 7.7](#), is at least $n - k' + 1$ by [Lemma 7.4](#).

Theorem 7.8

Let C be a k' -dimensional GRS code of length n . If the number of errors that occur in the transmission of a codeword of the alternant code $A(C)$ is at most $\lfloor \frac{1}{2}(n - k') \rfloor$, then there exists a polynomial time decoding algorithm which corrects the errors.

Proof

This is similar to the proof of [Theorem 6.6](#).

We suppose g is an arbitrary polynomial of degree $\lceil \frac{1}{2}(n + k') \rceil - 1$ and that h is an arbitrary polynomial of degree $\lfloor \frac{1}{2}(n - k') \rfloor$.

We determine the coefficients of g and h by solving the system of n equations,

$$g(a_j) - h(a_j)v_j^{-1}y_j = 0,$$

for $j = 1, \dots, n$. The homogeneous linear system has $n + 1$ unknowns (the coefficients of g and h) and n equations. Hence, we can find a non-trivial solution (in polynomial time) which determines $h(X)$ and $g(X)$.

By assumption, there is a polynomial f of degree at most $k' - 1$, such that $y_j = v_j f(a_j)$ for at least $n - \lfloor \frac{1}{2}(n - k') \rfloor$ values of j . For these values of j , the evaluation at a_j of

$$g(X) - h(X)f(X)$$

is zero. The degree of this polynomial is at most $\lceil \frac{1}{2}(n + k') \rceil - 1$. Since

$$n - \lfloor \frac{1}{2}(n - k') \rfloor > \lceil \frac{1}{2}(n + k') \rceil - 1,$$

it has more zeros than its degree, so it is identically zero. Therefore, $h(X)$ divides $g(X)$ and the quotient is $f(X) = g(X)/h(X)$. \square

Example 7.9

Suppose that we are using the code $A(C)$, where C is defined as in [Example 7.3](#), and that we have received the vector

$$y = (0, 0, 1, 0, 1, 0, 1, 1).$$

According to [Theorem 7.8](#), we should solve the equations

$$v_j g(a_j) - h(a_j) y_j = 0,$$

where $j = 1, \dots, 8$, and where $g(X)$ is a polynomial of degree at most 6 and

$$h(X) = h_1 X + h_0.$$

The equations are

$$g(0) = 0, \quad g(1) = 0, \quad g(e) = h(e), \quad g(e^2) = 0, \quad g(e^3) = h(e^3), \quad g(e^4) = 0,$$

$$g(e^5)e^5 = h(e^3), \quad g(e^6)e^3 = h(e^6).$$

From these equations we have that

$$g(X) = X(X + 1)(X + e^2)(X + e^4)(g_2 X^2 + g_1 X + g_0)$$

for some g_0, g_1 and g_2 and that

$$\begin{pmatrix} e^5 & e^4 & e^3 & e & 1 \\ 1 & e^4 & e & e^3 & 1 \\ e^6 & e & e^3 & e^5 & 1 \\ e^5 & e^6 & 1 & e^6 & 1 \end{pmatrix} \begin{pmatrix} g_2 \\ g_1 \\ g_0 \\ h_1 \\ h_0 \end{pmatrix} = 0.$$

This homogeneous system of equations has a solution with $g_2 = 1$ which gives

$$g(X) = X(X+1)(X+e^2)(X+e^4)(X^2+eX+e^2)$$

and

$$h(X) = eX + e^3.$$

Therefore, by [Theorem 7.8](#), the sent codeword is the evaluation of

$$\frac{g(X)}{h(X)} = e^6 X(X+1)(X+e^4)(X^2+eX+e^2).$$

One can check that the evaluation of this polynomial is the codeword

$$(0, 0, 1, 1, 1, 0, 1, 1). \quad \blacksquare$$

We can use the list decoding algorithm of [Theorem 6.9](#) to correct more distorted codewords. To do this we make a slight modification of the interpolation and solve the system of equations

$$Q(x_\ell, v_\ell^{-1}y_\ell) = 0.$$

This will produce a short list of possibilities for the sent codeword. The list of possibilities is further reduced since we can discard any vectors in the list which are not codewords of $A(C)$.

7.4 Algebraic Geometric Codes

Let ϕ be an absolutely irreducible homogeneous polynomial in $\mathbb{F}_q[X, Y, Z]$ of degree m . Recall that absolutely irreducible means that ϕ is irreducible over $\overline{\mathbb{F}}_q$, an algebraic closure of \mathbb{F}_q .

Let χ be the plane curve, defined as the points where ϕ is zero, where the points are points of the projective plane defined over $\overline{\mathbb{F}}_q$. It is not necessary to take χ to be a plane curve, but it will make things simpler and more apparent if we assume for the moment that it is. In the next section, we will consider an example of a higher dimensional curve. We direct the reader to the comments section for references to a more general treatment of algebraic geometric codes. Although we will define our codes over \mathbb{F}_q , we have to consider the curve over $\overline{\mathbb{F}}_q$. It is essential that when we apply Bezout's theorem, the number of points in the intersection of a curve defined by a homogeneous polynomial g and a curve defined by a homogeneous polynomial h is $(\deg g)(\deg h)$, where we have to count the intersections with multiplicity.

The **coordinate ring** is defined as the quotient ring

$$\mathbb{F}_q[\chi] = \mathbb{F}_q[X, Y, Z]/(\phi).$$

Therefore, the elements of $\mathbb{F}_q[\chi]$ are residue classes which can be represented by polynomials. We will only be interested in residue classes represented by homogeneous polynomials. We will be particularly interested in the elements of

$$\mathbb{F}_q(\chi) = \{f \mid f = g/h, \text{ for some homogeneous } g, h \in \mathbb{F}_q[\chi] \text{ of the same degree}\}.$$

An element f of $\mathbb{F}_q(\chi)$ can have very different representations.

Example 7.10

Let χ be the curve defined by $X^3 = Y^2Z$. The element f of $\mathbb{F}_q(\chi)$ represented by X^2/Y^2 is the same as the element of $\mathbb{F}_q(\chi)$ represented by Z/X . ■

The elements of $\mathbb{F}_q(\chi)$ do not in general define functions on the curve χ , since there may be a point P of χ where h is zero. However, there may be another representation of the same element of $\mathbb{F}_q(\chi)$, where h is not zero, so the evaluation of f is defined.

As in ▶ Section 2.4, we denote by $(x : y : z)$ the point of $\text{PG}(2, q)$ with vector representative (x, y, z) .

Example 7.11

In Example 7.10, f is defined at the point $P = (0 : 1 : 0)$, even though in the representation Z/X we have a zero in the denominator. Indeed, using the representation X^2/Y^2 , we deduce that f has a zero at P . However, f is not defined at the point $(0 : 0 : 1)$, where it has a pole. ■

The elements of \mathbb{F}_q are representatives of elements of $\mathbb{F}_q(\chi)$, since they are polynomials of degree 0 divided by a polynomial of degree 0. These elements define a constant function on the curve χ .

We define a **divisor** as a finite sum of the points of ϕ with integer coefficients,

$$D = \sum_{P \in \chi} n_P P.$$

At first glance, this seems like an odd thing to define. But it helps us keep track of the zeros and poles of an element of $\mathbb{F}_q(\chi)$ and this will be of utmost importance to us.

Assume that χ is a non-singular curve.

We define the divisor of $f \in \mathbb{F}_q(\chi)$ to be the divisor, denoted (f) , where we sum the zeros of g intersect ϕ , counted with multiplicity, and subtract the zeros of h intersect ϕ , counted with multiplicity. This definition of (f) is well-defined in that it does not depend on the representatives g and h for $f = g/h$.

The **degree** of a divisor D is

$$\deg(D) = \sum_{P \in \chi} n_P.$$

Bezout's theorem, as mentioned before, ensures that the degree of the divisor (f) , for any $f \in \mathbb{F}_q(\chi)$, is zero.

Example 7.12

Let us calculate the divisor of f , from Example 7.11, using the representative Z/X . The curve defined by the equation $Z = 0$ and the curve defined by $X^3 = Y^2Z$ intersect in the point $P_2 = (0 : 1 : 0)$ with multiplicity three. The curve defined by $X = 0$ and the curve defined by $X^3 = Y^2Z$ intersect in the point $P_2 = (0 : 1 : 0)$ and the point $P_3 = (0 : 0 : 1)$ with multiplicity two. Hence,

$$(f) = 3P_2 - 2P_3 - P_2 = 2P_2 - 2P_3.$$

If we had used the representative X^2/Y^2 , we would have arrived at the same conclusion. ■

Let

$$D = \sum_{P \in \chi} n_P P.$$

We write $D \geq 0$ if and only if the coefficients $n_P \geq 0$ for all $P \in \chi$.

Observe that the coefficients of (f) are positive and negative unless f is represented by a polynomial of degree zero. Hence, $(f) \geq 0$ if and only if $f = c$, for some $c \in \mathbb{F}_q$.

If P is a point of χ whose coordinates are all in \mathbb{F}_q , then we say that P is an \mathbb{F}_q -rational point of χ .

Suppose that D is a divisor in which the sum is restricted to \mathbb{F}_q -rational points. It is straightforward to verify that the subset

$$L(D) = \{f \in \mathbb{F}_q(\chi) \mid (f) + D \geq 0\}$$

is a vector space over \mathbb{F}_q . Moreover, its dimension can be calculated from the Riemann–Roch theorem. Precisely, if $\deg(D) \geq 2g - 1$, then

$$\dim L(D) = \deg(D) - g + 1,$$

where g is the genus of the curve χ . Recall that for a non-singular plane curve the genus of χ is $(m - 1)(m - 2)/2$, where m is the degree of ϕ .

Observe that if $\deg D < 0$, then $L(D) = \{0\}$.

We are now in a position to prove that the code whose codewords are the evaluation of the functions in $L(D)$ at certain points of the curve χ , will be a linear code over \mathbb{F}_q , whose dimension we know and whose minimum distance we can bound from below. In other words, we have a prescribed minimum distance as we did for BCH codes.

Theorem 7.13

Suppose

$$D = \sum_{P \in \chi} n_P P,$$

where the sum is over the \mathbb{F}_q -rational points of χ .

Let $\{P_1, \dots, P_n\}$ be a set of n \mathbb{F}_q -rational points of χ for which $n_{P_j} = 0$, for all $j \in \{1, \dots, n\}$.

If $n > \deg(D) \geq 2g - 1$, then

$$C(\chi, D) = \{(f(P_1), \dots, f(P_n)) \mid f \in L(D)\}$$

is a

$$[n, \deg(D) - g + 1, \geq n - \deg(D)]_q$$

code.

Proof

Let $\alpha : L(D) \rightarrow \mathbb{F}_q^n$ be defined by

$$\alpha(f) = (f(P_1), \dots, f(P_n)).$$

Since $L(D)$ is a vector space, this defines a linear map.

Let $E = P_1 + \dots + P_n$. If $f \in \ker \alpha$, then $f \in L(D - E)$, since $f \in L(D)$ and f is zero at P_1, \dots, P_n . Since $\deg(D - E) < 0$, this implies $f = 0$. Therefore, the image of α has dimension $\dim L(D)$, which is $\deg(D) - g + 1$, by the Riemann–Roch theorem mentioned above.

Suppose $\alpha(f)$ has weight $w > 0$. Then, after a suitable reordering of the points, we can assume

$$f(P_1) = \dots = f(P_{n-w}) = 0.$$

Since $f \neq 0$ and $f \in L(D - P_1 - \dots - P_{n-w})$, this implies that

$$\deg(D - P_1 - \dots - P_{n-w}) \geq 0,$$

which gives $\deg(D) \geq n - w$. Therefore, the minimum weight of a non-zero codeword of C is at least $n - \deg(D)$ which, by Lemma 4.1, implies that the minimum distance of the linear code $C(\chi, D)$ is at least $n - \deg(D)$. \square

Example 7.14

Let χ be the curve of genus 1 defined as the zeros of the polynomial

$$X^3 + Y^2Z + Z^2Y \in \mathbb{F}_4[X, Y, Z].$$

The line $X = 0$ intersects χ in the points $Q = (0 : 1 : 0)$, $P = (0 : 0 : 1)$ and $R = (0 : 1 : 1)$.

The line $Y = 0$ intersects χ in the point P with multiplicity 3 and the line $Z = 0$ intersects χ in the point Q with multiplicity 3.

Suppose $D = 3Q$. Then, since

$$\left(\frac{X}{Z}\right) = P + R - 2Q$$

and

$$\left(\frac{Y}{Z}\right) = 3P - 3Q$$

and

$$\dim L(D) = \deg(D) - g + 1 = 3,$$

we have that

$$\left\{\frac{X}{Z}, \frac{Y}{Z}, 1\right\}$$

is a basis for $L(D)$. A generator matrix for $C(\chi, 3Q)$ is given by

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

where $e \in \mathbb{F}_4$ and $e^2 = e + 1$. In this case we get a generator matrix whose columns are the eight \mathbb{F}_4 -rational points of χ not equal to Q .

By Theorem 7.13, the minimum distance of $C(\chi, 3Q)$ is at least 5. The codeword $(1, 1, 1)G$ has weight 5, so $C(\chi, 3Q)$ is an $[8, 3, 5]_4$ code.

Now, suppose $D = 5Q$. By Theorem 7.13 the dimension of $C(\chi, 5Q)$ is 5. To find a generator matrix for $C(\chi, 5Q)$, we can extend the basis for $L(3Q)$ to $L(5Q)$ by observing that

$$\left(\frac{X^2}{Z^2}\right) = 2P + 2R - 4Q$$

and

$$\left(\frac{XY}{Z^2}\right) = 4P + R - 5Q.$$

Therefore,

$$\left\{\frac{X}{Z}, \frac{Y}{Z}, \frac{X^2}{Z^2}, \frac{XY}{Z^2}, 1\right\}$$

is a basis for $L(5Q)$.

Thus, $C(\chi, 5Q)$ has a generator matrix

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

According to [Theorem 7.13](#), the minimum distance of $C(\chi, 5Q)$ is at least $n - \deg(5Q) = 3$. The codeword $(1, 1, 1, 1, 0)G'$ has weight 3, so $C(\chi, 5Q)$ has minimum distance equal to 3. Therefore, $C(\chi, 5Q)$ is an $[8, 5, 3]_4$ code. ■

7.5 Algebraic Geometric Codes Surpassing the Gilbert–Varshamov Bound

Algebraic geometric codes are of particular interest because they can provide examples of asymptotically good codes which better the Gilbert–Varshamov bound for certain large enough alphabets. We first calculate the asymptotic Gilbert–Varshamov bound for a general alphabet of size r , as we did in [Corollary 3.8](#) for binary codes.

Supposing that the codes have rate R and relative minimum distance δ , the bound in [Theorem 3.7](#) gives

$$\binom{n}{\delta n} (r-1)^{\delta n} r^{nR} > r^n,$$

which by [Lemma 1.11](#) gives

$$2^{h(\delta)n} (r-1)^{\delta n} r^{nR} > r^n.$$

Taking logarithms and dividing by n , we have that the r -ary asymptotic Gilbert–Varshamov bound is

$$R > 1 - h(\delta) \log_r 2 - \delta \log_r (r - 1).$$

Theorem 7.13 implies that a k -dimensional algebraic geometric code of a curve of genus g satisfies

$$k + d \geq n - g + 1,$$

where d is the minimum distance and n is the length, which is maximised when we take as many \mathbb{F}_q -rational points of χ as possible. Dividing by n , this gives the bound

$$R + \delta \geq 1 - g/n + 1/n.$$

Therefore, to find asymptotically good codes, we need a sequence of curves C_i ($i \in \mathbb{N}$) of genus g_i , for which $g_i \rightarrow \infty$, and where g_i/n_i tends to a number smaller than 1. Here, n_i is the number of \mathbb{F}_q -rational points of C_i . One way to construct such curves is with recursive towers. These curves are constructed from an absolutely irreducible polynomial $f \in \mathbb{F}_q[X, Y]$. The affine points of C_i are defined in the following way. The curve C_1 is defined by $f(X_1, X_2) = 0$. The second curve C_2 is defined by

$$f(X_1, X_2) = f(X_2, X_3) = 0,$$

the third curve C_3 is defined by

$$f(X_1, X_2) = f(X_2, X_3) = f(X_3, X_4) = 0,$$

and so on. Observe that the curve C_i is a set of points in $\text{PG}(i + 1, q)$, although we have only described the affine part of the curve. The points on the hyperplane at infinity are obtained by homogenising the polynomials and setting the new variable to zero.

If q is an even power of an odd prime, then the sequence of curves constructed in this way from

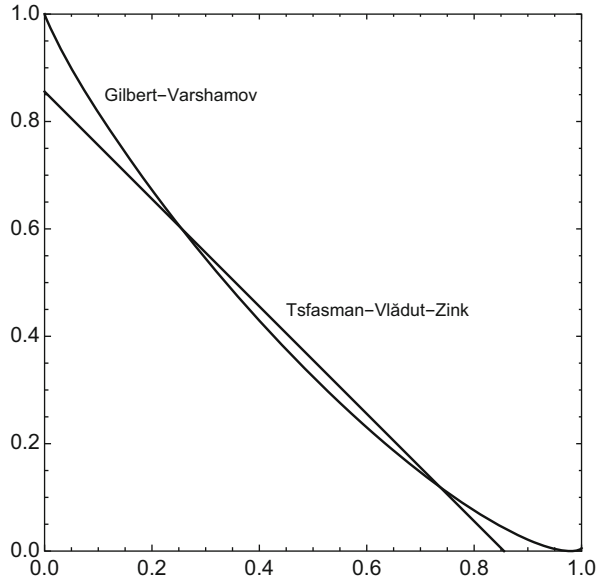
$$f(X, Y) = (Y^{\sqrt{q}} + Y)(1 + X^{\sqrt{q}-1}) - X^{\sqrt{q}}$$

is a sequence of asymptotically good codes of rate R and relative minimum distance δ for which

$$R + \delta \geq 1 - (\sqrt{q} - 1)^{-1}.$$

For $q \geq 49$, this line surpasses the Gilbert–Varshamov bound, for some range of values of δ , see [Figure 7.1](#). This example is due to Tsfasman, Vlăduț and Zink.

Fig. 7.1 The asymptotic Gilbert–Varshamov curve and the Tsfasman–Vlăduț–Zink bound for $q = 49$.



7.6 Comments

Alternant codes were introduced by Helgert [36] in 1974, Goppa having previously considered a subclass of alternant codes [28]. It was Goppa's later work from [29] which led to algebraic geometric codes. The book of Tsfasman and Vlăduț on algebraic geometric codes [72] was published in 1991 and contains a wealth of results on such codes. The particular bound in Figure 7.1 is from [73] and a survey of the asymptotic bounds can be found in Tsfasman [71].

7.7 Exercises

7.1 Suppose that we have received the vector $(-1, 1, 1, 1, 0, 0, 1, 1, 1)$, having sent a codeword of the subfield subcode in Example 7.2. Use syndrome decoding to find and correct the error bit.

7.2 Let M be an invertible matrix and let H be as in Exercise 7.6. Let C be the linear code over \mathbb{F}_{q^h} with check matrix H and let C' be the linear code over \mathbb{F}_{q^h} with check matrix MH . Prove that $A(C) = A(C')$.

7.3 Let $\phi(X)$ be a polynomial in $\mathbb{F}_{q^h}[X]$ of degree r and let $\{\alpha_1, \dots, \alpha_n\}$ be the set of elements of \mathbb{F}_{q^h} which are not zeros of $\phi(X)$. For each $u = (u_1, \dots, u_n) \in \mathbb{F}_{q^h}^n$, let

$$f_u(X) = \sum_{i=1}^n u_i \left(\frac{\phi(\alpha_i) - \phi(X)}{X - \alpha_i} \right) \phi(\alpha_i)^{-1}.$$

Prove that the subfield subcode $A(C)$, where

$$C = \{(u_1, \dots, u_n) \in \mathbb{F}_{q^h} \mid f_u(X) = 0\},$$

is an alternant code.

7.4 Let C be the linear code over \mathbb{F}_8 with check matrix

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ e & e^2 & e^3 & e^4 & e^5 & e^6 & e^7 & 0 \end{pmatrix},$$

where e is a primitive element of \mathbb{F}_8 .

Prove that the binary alternant code $A(C)$ is a $[8, 4, 4]_2$ code.

7.5 Let e be a primitive element of \mathbb{F}_8 , such that $e^3 = e + 1$ and let C be the GRS code

$$\{(e^4 f(0), f(1), e^6 f(e), e^4 f(e^2), f(e^3), e^6 f(e^4), e^4 f(e^5), f(e^6)) \mid f \in \mathbb{F}_8[X], \deg f \leq 4\}.$$

Prove that

$$A(C) = \{(0, 0, 0, 0, 0, 0, 0, 0), (1, 1, 1, 1, 0, 1, 0, 1)\}.$$

7.6

i. Prove that the GRS code in [Lemma 7.4](#) has a check matrix

$$H = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \alpha_1^{n-k'-1} & \alpha_2^{n-k'-1} & \dots & \alpha_n^{n-k'-1} \end{pmatrix} \begin{pmatrix} v_1^{-1} & 0 & \dots & 0 \\ 0 & v_2^{-1} & 0 & \dots \\ 0 & 0 & \ddots & 0 \\ \vdots & \dots & 0 & \ddots \\ 0 & \dots & \dots & 0 & v_n^{-1} \end{pmatrix}.$$

ii. Prove that the dual of a GRS code is a GRS code.

7.7 Let C be the GRS code defined over $\mathbb{F}_4 = \{0, 1, e, e^2\}$ by

$$C = \{(f(0), f(1), e^{-1} f(e), e^{-2} f(e^2)) \mid f \in \mathbb{F}_4[X], \deg f \leq 1\}.$$

Suppose that we have received the vector $y = (1, 1, 1, 1)$. Use the decoding algorithm in [Theorem 7.8](#) to find g and h , polynomials of degree 2 and 1, respectively. Verify that h divides g , deduce f and correct the error in y .

7.8 Prove that, in the decoding algorithm of [Theorem 7.8](#), if $n + k'$ is odd, then

$$\deg g \leq \lceil \frac{1}{2}(n + k') \rceil - 2.$$

7.9 Let C be the GRS code defined over \mathbb{F}_9 by

$$C = \{(f(0), f(1), ef(e), f(e^2), e^2 f(e^3)) \mid f \in \mathbb{F}_9[X], \deg f \leq 1\},$$

where $e^2 = e + 1$.

Suppose that we have received the vector $y = (1, 2, 1 + 2e, 2 + e, 0)$.

Use the decoding algorithm in [Theorem 7.8](#) to find g and h , polynomials of degree at most 3 and 1, respectively. Verify, as claimed in [Exercise 7.8](#), that the degree of g is 2, that h divides g , deduce f and correct the error in y .

7.10 Let χ be the curve defined as the zeros of the polynomial

$$X^4 + Y^3Z + Z^3Y.$$

Let $P_1 = (1 : 0 : 0)$, $P_2 = (0 : 1 : 0)$ and $P_3 = (0 : 0 : 1)$.

- i. Calculate the divisor (Y/Z) .
- ii. Find a basis for $L(D)$, where $D = 3P_1 + 3P_2$.
- iii. The curve χ has 28 rational points over \mathbb{F}_9 . Construct a $[24, 4, 18]_9$ code.
- iv. Verify that the Griesmer bound for a $[24, 4, d]_9$ code gives $d \leq 19$.



Low Density Parity Check Codes

A linear code with a check matrix in which each column has few non-zero entries is called a low density parity check code or, for brevity, an LDPC code. These codes were introduced in the 1960s by Gallager who proved that probabilistic constructions of such matrices produce asymptotically good linear codes. Moreover, he observed that LDPC codes perform well when applying the following decoding algorithm. On receiving a vector v , one calculates the weight of the syndrome of $v + e$, for each vector e of weight one. If the weight of this syndrome is less than the weight of the syndrome of v , for some e , then we replace v by $v + e$ and repeat the process. If at each iteration there is such a vector e , then, since after replacing v by $v + e$, the weight of the syndrome of v decreases, we will eventually find a vector whose syndrome is zero, which must be the syndrome of some codeword u . We then decode v as u . If at some iteration no such e exists then the decoding breaks down. If at some iteration more than one such vector e exists, then one could choose e so that the weight of the syndrome of $v + e$ is minimised. In this chapter we will prove that there are LDPC codes, constructed from graphs with the expander property, for which the decoding algorithm will not break down. Provided that the number of error bits is less than half the minimum distance, the decoding algorithm will return the nearest codeword to the received vector. We will use probabilistic arguments to construct the graphs, and from these a sequence of codes which are asymptotically good.

8.1 Bipartite Graphs with the Expander Property

A **graph** is a pair (V, E) , where V is a set and E is a set of 2-subsets of V . We consider the elements of V to be vertices and the set E to be a set of edges, where an edge $\{v_1, v_2\}$ joins the two vertices v_1 and v_2 . A **bipartite graph** is a graph in which V is the disjoint union $V_1 \cup V_2$ and for all $e = \{v_1, v_2\} \in E$, we have $v_1 \in V_1$ and $v_2 \in V_2$. In other words, there are no edges joining two vertices in V_1 and no edges joining two vertices in V_2 . The subsets V_1 and V_2 are called the **stable sets** of the bipartite graph. The **degree** of a vertex v is the number of edges which contain v and we say that u is a **neighbour** of v if $\{u, v\}$ is an edge. A bipartite graph with stable sets V_1 and V_2 is **left γ -regular** if

every vertex in V_1 has degree γ , i.e. has γ neighbours in V_2 . For a subset S of V_1 , denote by $N(S)$ the set of vertices of V_2 which are neighbour to some vertex in S .

Let δ be a real number in the interval $(0, 1)$. A left γ -regular bipartite graph with stable sets V_1 and V_2 of size n and m , respectively, has the **expander property** with respect to δ if for all subsets S of V_1 of size less than δn , $|N(S)| > \frac{3}{4}\gamma|S|$.

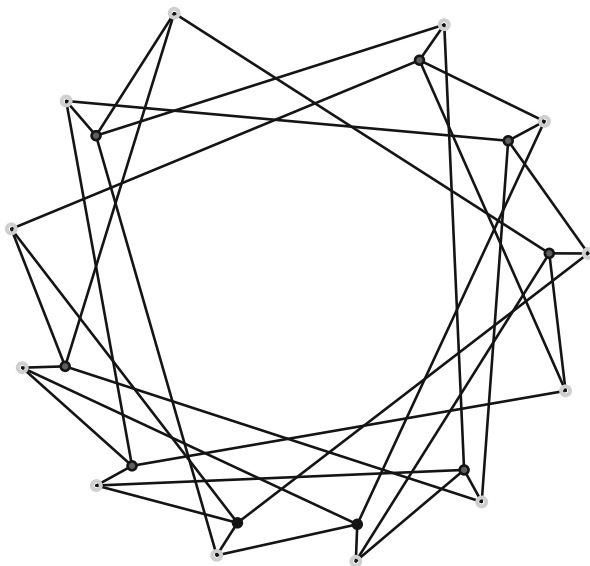
Example 8.1

■ **Figure 8.1** is a left 3-regular bipartite graph with the expander property with respect to $\delta = \frac{1}{4}$. The 12 vertices of V_1 are the vertices on the outer circle and the 9 vertices of V_2 are the vertices on the inner circle. One can verify that every subset S of V_1 of size 1 has three neighbours, so the graph is left 3-regular. If S is a 2-subset of V_1 , then $N(S)$ has either 5 or 6 vertices. Therefore, for every subset S of V_1 of size less than $\delta n = 3$, $|N(S)| > \frac{9}{4}|S|$. ■

Lemma 8.2 Given $\gamma > 4$ and $R \in (0, 1)$, there is a constant $\delta \in (0, 1)$, dependent on γ and R , for which a left γ -regular bipartite graph with the expander property with respect to δ exists, for all n large enough, where the left and right stable sets have size n and $\lfloor (1 - R)n \rfloor$, respectively.

Proof

Consider the set Θ of bipartite left γ -regular graphs with stable sets V_1 and V_2 of size n and $m = \lfloor (1 - R)n \rfloor$.



■ **Fig. 8.1** A left 3-regular bipartite graph with the expander property.

8.1 · Bipartite Graphs with the Expander Property

For a graph $\Gamma \in \Theta$, a subset S of V_1 of size $s < \delta n$ and a subset T of V_2 of size $\lfloor \frac{3}{4}\gamma s \rfloor$, define a random variable $X_{S,T}$ which takes the value 1 if all edges of Γ with an end-vertex in S have an end-vertex in T and 0 otherwise. If we can prove that the probability

$$P\left(\sum_{S,T} X_{S,T} = 0\right) \neq 0,$$

then we can deduce that there is a graph $\Gamma \in \Theta$ for which $X_{S,T} = 0$ for all subsets S and T . This implies that the graph Γ has the property that the union of the neighbours of the vertices in S has more than $\frac{3}{4}\gamma|S|$ vertices, for all subsets S of V_1 of size less than δn . Hence, Γ has the expander property with respect to δ .

Since the probability that a randomly chosen edge has an end-vertex in T is $\lfloor \frac{3}{4}\gamma s \rfloor / m$,

$$\sum_{S,T} P(X_{S,T} = 1) < \sum_{s=1}^{\lfloor \delta n \rfloor} \binom{n}{s} \binom{m}{\lfloor \frac{3}{4}\gamma s \rfloor} \left(\frac{\lfloor \frac{3}{4}\gamma s \rfloor}{m}\right)^{\gamma s},$$

where the sum is over all subsets S of V_1 of size $s < \delta n$ and all subsets T of V_2 of size $\lfloor \frac{3}{4}\gamma s \rfloor$.

Since

$$e^k = \sum_{j=0}^{\infty} \frac{k^j}{j!} > \frac{k^k}{k!},$$

we have

$$\binom{n}{k} \leq \frac{n^k}{k!} < \left(\frac{ne}{k}\right)^k,$$

which in the above gives

$$\sum_{S,T} P(X_{S,T} = 1) < \sum_{s=1}^{\delta n} \left(\frac{en}{s}\right)^s \left(\frac{e\lfloor(1-R)n\rfloor}{\lfloor \frac{3}{4}\gamma s \rfloor}\right)^{\frac{3}{4}\gamma s} \left(\frac{\lfloor \frac{3}{4}\gamma s \rfloor}{\lfloor(1-R)n\rfloor}\right)^{\gamma s}.$$

Since

$$s^{(\frac{1}{4}\gamma-1)s} < (\delta n)^{(\frac{1}{4}\gamma-1)s},$$

this gives

$$\sum_{S,T} P(X_{S,T} = 1) < \sum_{s=1}^{\delta n} (N\delta^{\frac{1}{4}\gamma-1})^s,$$

for some constant N , dependent on γ and R , but not dependent on n .

Now, if we choose δ so that $N\delta^{\frac{1}{4}\gamma-1} < \frac{1}{2}$, then this sum is less than 1.

Since,

$$P\left(\sum_{S,T} X_{S,T} \neq 0\right) \leq \sum_{S,T} P(X_{S,T} = 1) < 1,$$

we have that

$$P\left(\sum_{S,T} X_{S,T} = 0\right) \neq 0.$$

□

8.2 Low Density Parity Check Codes

A **low density parity check code** is a linear code which has a check matrix H with the property that H has few non-zero elements in each column. We will only consider low density parity check binary codes, so H will have the property that it has few 1's in each column. We make this vague definition precise for sequences of codes of length n , where n tends to infinity, by insisting that the few non-zero elements is a constant number.

Lemma 8.2 proves that sequences of bipartite expander graphs exist for n large enough. We now prove that we can construct a matrix from a bipartite graph in this sequence, which is a check matrix of a code in a sequence of asymptotically good linear codes.

Fix R to be the rate of transmission we would like to achieve and δ to be the relative minimum distance.

Lemma 8.3 *Given a left γ -regular bipartite graph Γ with stable sets of size n and $m = \lfloor (1 - R)n \rfloor$ and the expander property with respect to δ , there exists a binary linear code $C(\Gamma)$ with rate at least R and relative minimum distance at least δ .*

Proof

Let H be the $m \times n$ matrix whose rows are indexed by the vertices of V_2 and whose columns are indexed by the vertices of V_1 . A row-column entry is 1 if there is an edge joining the vertex of V_1 to the vertex of V_2 and zero otherwise. Then H has γ 1's in each column.

Let $C(\Gamma)$ be the binary linear code defined by the check matrix H , i.e.

$$C(\Gamma) = \{u \in \mathbb{F}_2^n \mid uH^t = 0\}.$$

Since the rank of H is at most the number of rows, the dimension of $C(\Gamma)$ is at least $n - m = n - \lfloor (1 - R)n \rfloor \geq Rn$, so $C(\Gamma)$ will have rate at least R .

Suppose that $C(\Gamma)$ has minimum distance less than δn . By **Lemma 4.1**, there is a non-zero vector u of $C(\Gamma)$ of weight less than δn . Let S be the support of u , the set of coordinates where u has a 1. These coordinates correspond to vertices of V_1 , since the rows of H^t are

indexed by the vertices of V_1 , so we can think of S as a subset of V_1 . Since $|S| < \delta n$ and Γ has the expander property with respect to δ , the size of the set $N(S)$, the set of vertices neighbour to some vertex of S , is at least $\frac{3}{4}\gamma|S|$.

If every vertex of $N(S)$ has at least two edges joining it to vertices of S , then, counting edges with an end-vertex in S ,

$$|S|\gamma \geq 2|N(S)|, \quad (8.1)$$

which contradicts $|N(S)| \geq \frac{3}{4}\gamma|S|$.

Therefore, there is some vertex v in $N(S)$ which is joined to just one vertex of S . A vertex of V_2 indexes a row r of the check matrix H . Since r is a row of the check matrix, it has the property that $r \cdot w = 0$, for all codewords $w \in C(\Gamma)$.

However, v is joined to just one vertex of the support S of u , so $r \cdot u = 1$, which is a contradiction, since the scalar product of a row of the check matrix H and a codeword is zero. Hence, each non-zero vector in $C(\Gamma)$ has weight at least δn . By [Lemma 4.1](#), $C(\Gamma)$ has minimum distance at least δn . □

Observe that we could relax the expander property to $|N(S)| > \frac{1}{2}\gamma|S|$ and [Lemma 8.3](#) would still hold. However, we insist upon $|N(S)| > \frac{3}{4}\gamma|S|$, so that the decoding algorithm, which we will see in the following section, works.

Example 8.4

The check matrix obtained from the bipartite graph Γ in [Example 8.1](#) is

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

[Lemma 8.3](#) implies that the rate of $C(\Gamma)$ is at least $\frac{1}{4}$ and the minimum distance is at least 3. It is [Exercise 8.1](#) to verify that H has rank 9 and so the dimension of $C(\Gamma)$ is 3 and the rate of $C(\Gamma)$ is precisely $\frac{1}{4}$.

Consider the graph in [Example 8.1](#). We can define a geometry which has as points the vertices of V_2 and as lines the vertices of V_1 , where for each vertex u of V_1 , we have a line of the geometry consisting of the points which are neighbours to u in the graph. This geometry is $AG(2, 3)$, the affine plane of order 3, see [Section 2.4](#), and in particular [Figure 2.2](#).

Each block of three columns corresponds to a parallel set of lines, so $C(\Gamma)$ has a generator matrix

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

Thus we deduce that all codewords of $C(\Gamma)$ have weight 0, 6 or 12, so $C(\Gamma)$ is a $[12, 3, 6]_2$ code.

The Griesmer bound from [Theorem 4.18](#) for a $[12, 3, d]_2$ code gives the bound $d \leq 6$. ■

8.3 Decoding LDPC Codes

[Lemma 8.2](#) and [Lemma 8.3](#) imply that we can find asymptotically good codes using bipartite graphs with the expander property. Moreover, these codes are linear, so fast to encode. However, the really useful property that these codes have is that they are also fast to decode. We will prove this in [Theorem 8.7](#), but first we need to prove the following lemma.

Recall that for $x \in \mathbb{F}_2^n$, the syndrome of x is

$$s(x) = xH^t$$

and $\text{wt}(x)$ is the number of non-zero coordinates that x has.

Let e_i denote the vector of weight one and length n with a 1 in the i -th coordinate.

Lemma 8.5 *Suppose that $x \in \mathbb{F}_2^n$ and that $d(x, u) < \delta n$ for some $u \in C(\Gamma)$, where $C(\Gamma)$ is the binary linear code obtained from a bipartite graph Γ which has the expander property with respect to δ . Then there is an $i \in \{1, \dots, n\}$ such that $\text{wt}(s(x + e_i)) < \text{wt}(s(x))$.*

Proof

Let S be the coordinates where x and u differ. By assumption, $|S| < \delta n$. As in the proof of [Lemma 8.3](#), the set S corresponds to a subset of the vertices V_1 of the graph Γ . As before, let $N(S)$ denote the set of vertices which are neighbour to some vertex of S . Then, as in the proof of [Lemma 8.3](#), the vertices of $N(S)$ index rows of the matrix H which in turn correspond to linear constraints on the code $C(\Gamma)$. Divide $N(S)$ into T , constraints that x satisfies and U , constraints that x does not satisfy. In other words, T is the subset of $N(S)$ where $s(x)$ has a zero and U is the subset of $N(S)$ where $s(x)$ has a 1. Here, we are identifying the coordinates of $s(x)$ with vertices of V_2 .

Since $|S| < \delta n$, the expander property implies

$$|T| + |U| = |N(S)| > \frac{3}{4}\gamma|S|.$$

Counting edges between S and $N(S)$, we have

$$|U| + 2|T| \leq \gamma|S|,$$

since there must be at least two edges joining T to vertices in S , otherwise the constraint would not be satisfied.

Combining the inequalities this implies $|T| < \frac{1}{4}\gamma|S|$ and therefore

$$|U| > \frac{1}{2}\gamma|S|. \quad (8.2)$$

This implies that

$$\text{wt}(s(x)) > \frac{1}{2}\gamma|S|.$$

Since,

$$s(x) = s(x + u) = \sum_{i \in S} s(e_i),$$

the pigeon-hole principle implies that there is an i for which $s(e_i)$ and $s(x)$ both have a 1 in more than $\frac{1}{2}\gamma$ of the coordinates. Since $\text{wt}(s(e_i)) = \gamma$, this implies that

$$\text{wt}(s(x + e_i)) < \text{wt}(s(x)).$$

□

Example 8.6

Suppose that we have sent a codeword of $C(\Gamma)$, defined by the check matrix in [Example 8.4](#), and that we have received

$$x = (0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0).$$

By calculating xH^t ,

$$s(x) = (0, 0, 0, 1, 1, 1, 1, 1, 1, 1).$$

The weights of $s(x + e_i)$ are given in the following table:

i	1	2	3	4	5	6	7	8	9	10	11	12
$\text{wt}(s(x + e_i))$	5	5	5	5	5	5	9	3	3	5	5	5.

Assuming that less than two errors have occurred in transmission, [Lemma 8.5](#) implies that there is an i for which $\text{wt}(s(x + e_i)) < 6$ and this is the case for all $i \neq 7$.

We will return to this example in [Example 8.8](#). ■

[Lemma 8.5](#) allows us to apply the decoding algorithm described in [Theorem 8.7](#). This type of decoding algorithm is called **belief propagation**. Observe that in [Lemma 8.5](#) we are not claiming that by summing e_i to x we are correcting an error,

only that the weight of the syndrome decreases. It may be that in the algorithm in [Theorem 8.7](#), we introduce new errors at a particular iteration. However, since the weight of the syndrome is decreasing, it will eventually have weight zero and all errors, even those which we may have inadvertently introduced, will have been corrected.

Theorem 8.7

Let $C(\Gamma)$ be the linear code of length n obtained from a bipartite graph Γ with the expander property with respect to δ . There is a decoding algorithm for $C(\Gamma)$ which completes in a number of steps which is polynomial in n and which corrects up to $\frac{1}{2}\delta n$ error bits.

Proof

We will provide an algorithm for decoding the received vector $x \in \mathbb{F}_2^n$, where $d(x, u) < \frac{1}{2}\delta n$, for some $u \in C$.

Let S be the support of $x - u$, i.e. the coordinates where x and u differ.

Although we do not know what S is, by assumption $|S| < \frac{1}{2}\delta n$.

By [Lemma 8.5](#), if we test $x + e_i$ for $i = 1, \dots, n$, we will find an i for which

$$\text{wt}(s(x + e_i)) < \text{wt}(s(x)).$$

So, we can repeat this process with $x + e_i$ and $|U|$, the size of the set of unsatisfied constraints, will decrease.

The maximum value of $|U|$ (at the first step since $|U|$ is decreasing) is bounded by

$$|U| \leq |N(S)| \leq \gamma|S| < \frac{1}{2}\gamma\delta n.$$

At each iteration, when we apply [Lemma 8.5](#), equation (8.2) implies $|S| < \delta n$. Therefore, the hypothesis of [Lemma 8.5](#) is satisfied and can be applied at the next iteration.

The weight of $s(x)$ in the first iteration is less than, $\frac{1}{2}\gamma\delta n$. In each of the following iterations the weight of the syndrome decreases, so the number of iterations will be at most $\frac{1}{2}\gamma\delta n$. In each iteration we have to multiply a matrix with n vectors, so the whole algorithm completes in a number of steps which is polynomial in n . \square

Example 8.8

[Theorem 8.7](#) only guarantees that we can correct up to $\frac{1}{2}\delta n$ errors. In [Example 8.4](#), this implies that we can correct up to only one error bit even though the minimum distance is 6 and we would expect to be able to correct up to two error bits.

If we apply the algorithm described in [Theorem 8.7](#) to the table of syndromes we calculated in [Example 8.6](#), then we have many choices for i in the first iteration. Let us suppose we decide to choose an i for which the weight of $s(x + e_i)$ is minimised at each iteration. Then we would replace x by $x + e_8$ at the first iteration, then $x + e_8$ by $x + e_8 + e_9$ at the second iteration and would have correctly found the codeword

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

at distance 2 to x .



8.4 Comments

This chapter has realised an important aim that of proving that there are asymptotically good codes which we can encode and decode in an efficient manner. There are LDPC codes whose rate nears the Shannon capacity, see MacKay and Neal [49]. For an excellent survey on expanders, see Hoory, Linial and Wigderson [39]. The decoding algorithm for expander codes is due to Sipser and Spielman [67]. Regarding [Exercise 8.5](#), a list of the known hyperovals and their collineation groups can be found in Penttala and Pinneri's article [56]. For more on LDPC codes constructed from finite geometries, see Kou, Lin and Fossorlie [45] and Pepe [57].

LDPC codes have replaced **turbo codes** in 5G mobile networks. Turbo codes are not covered here in this text, principally because they lack any algebraic structure. LDPC codes are widely used in mobile and satellite communication and have been employed by NASA in recent missions as an alternative to Reed–Solomon codes.

8.5 Exercises

8.1 Prove that there is no subset S of points of $\text{AG}(2, 3)$ with the property that every line is incident with an even number of points of S . Conclude that the check matrix from [Example 8.4](#) has rank 9 over \mathbb{F}_2 .

8.2 Let Γ be the bipartite graph whose vertices V_1 are the lines of $\text{PG}(3, 4)$ and whose vertices V_2 are the points of $\text{PG}(3, 4)$ and where a point is joined to a line by an edge in Γ if and only if the point and line are incident in the geometry.

- i. Apply [Lemma 8.3](#) to prove that $C(\Gamma)$ is a linear code of rate at least $\frac{272}{357}$ with relative minimum distance at least $\frac{4}{357}$.
- ii. Prove that the minimum distance of $C(\Gamma)$ is at least 9.
- iii. Determine up to how many error bits we can guarantee to correct when applying belief propagation decoding.

8.3 Prove that the point-line incidence matrix of $\text{AG}(2, 4)$ is the adjacency matrix of a left 4-regular graph Γ with the expander property with respect to $\delta = \frac{1}{5}$, with stable sets of size 20 and 16.

8.4 Let q be odd.

- i. Prove that there is no set S of points of $\text{AG}(2, q)$ with the property that every line is incident with an even number of points of S .
- ii. Let H be the matrix whose rows are indexed by the points of $\text{AG}(2, q)$ and whose columns are indexed by the lines of $\text{AG}(2, q)$ and where an entry in the matrix is 1 if the point is incident with the line and 0 otherwise.

Let C be the binary linear code with check matrix H . Prove that the dimension of C is at least q and that the minimum distance of C is at least $q + 2$.

A **dual hyperoval** in $AG(2, q)$ or $PG(2, q)$ is a subset L of $q + 2$ lines with the property that every point is incident with either zero or two lines of L .

8.5

- i. Prove that if $AG(2, q)$ has a dual hyperoval, then q is even.
- ii. Suppose that u is a codeword of weight 6 of the binary linear code $C(\Gamma)$, where Γ is the expander graph in [Exercise 8.3](#). Prove that u is the indicator vector of a dual hyperoval. In other words, if S is the support of u , then, as a set of lines of $AG(2, 4)$, S is a dual hyperoval.
- iii. Deduce the minimum distance of $C(\Gamma)$ by proving that there are no small sets of lines L with the property that every point of $AG(2, 4)$ is incident with an even number of lines of L .
- iv. By calculating the rank of a check matrix for $C(\Gamma)$ (with the aid of a computer), determine the dimension of $C(\Gamma)$.



Reed–Muller and Kerdock Codes

In ► **Chapter 6**, we studied Reed–Solomon codes, codes whose codewords are the evaluation of polynomials in one variable of degree at most $k - 1$ at the elements of $\mathbb{F}_q \cup \{\infty\}$. Reed–Solomon codes are short length codes, where the length n is bounded by $q + 1$, and only useful when we take the field to be large. The alternant codes which we constructed from generalised Reed–Solomon codes in ► **Chapter 7** allowed us to construct codes over small fields and we put this to good use. In this chapter we will consider another generalisation of Reed–Solomon codes, codes whose codewords are the evaluation of polynomials in many variables. This again allows us to construct linear codes over small fields and we will restrict our attention, for the most part, to binary linear codes. It will turn out that these codes are not asymptotically good. Nevertheless, they are an important class of codes which are widely implemented due to the availability of fast decoding algorithms. One example of such a decoding algorithm is the majority-logic decoding algorithm that we will study here. We will then go on and construct Kerdock codes which are certain subcodes of the second-order Reed–Muller codes. These codes can give examples of non-linear codes with parameters for which no linear code exists.

9.1 Binary Reed–Muller Codes

A **Boolean function** from \mathbb{F}_2^m to \mathbb{F}_2 is the evaluation map of a polynomial with coefficients from \mathbb{F}_2 in m variables generated by monomials in which the degree of any particular indeterminate is at most 1.

Note that for both elements x of \mathbb{F}_2 , $x^2 = x$, so the function defined by the evaluation of the polynomial $x_1^2 x_2^3 x_3$ at the elements of \mathbb{F}_2^3 and the polynomial $x_1 x_2 x_3$ will be the same. Therefore, it makes sense that when considering evaluations of polynomials in many variables over \mathbb{F}_2 , we restrict our attention to Boolean functions.

The **r -th order Reed–Muller code** is a binary code $R(r, m)$ of length 2^m defined by

$$R(r, m) = \{(f(a_1), \dots, f(a_{2^m})) \mid \deg f \leq r\},$$

where $\{a_1, \dots, a_{2^m}\}$ is the set of vectors of \mathbb{F}_2^m and f runs through all Boolean functions that are defined by polynomials in m indeterminates of degree at most r .

The code $R(r, m)$ is a linear code over \mathbb{F}_2 , since

$$(f(a_1), \dots, f(a_{2^m})) + (g(a_1), \dots, g(a_{2^m})) = ((f + g)(a_1), \dots, (f + g)(a_{2^m})).$$

The vector space of Boolean functions of degree at most r in m variables has a canonical basis, which is the set of monomials of degree at most r in m variables and degree at most one in any particular variable. Therefore, the code $R(r, m)$ has a generator matrix whose rows are indexed by these monomials. For example, the set of monomials

$$\{1, x_1, \dots, x_m, x_1x_2, \dots, x_{m-1}x_m\}$$

is a basis for the vector space of Boolean functions in m variables of degree at most 2.

Example 9.1

The 11×16 matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{matrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_1x_2 \\ x_1x_3 \\ x_1x_4 \\ x_2x_3 \\ x_2x_4 \\ x_3x_4 \end{matrix}$$

is a generator matrix of the code $R(2, 4)$, with the rows being indexed by the monomials in four variables of degree at most two. ■

We have already proved the following lemma.

Lemma 9.2 $R(r, m)$ is a linear code of length 2^m and of dimension

$$1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r}.$$

Since $R(r, m)$ is linear, Lemma 4.1 implies that its minimum distance is equal to the minimum weight of a non-zero codeword. In the above example, evidently $R(2, 4)$ has codewords of weight 4 and this is indeed its minimum distance. In Theorem 9.3 we will

calculate the minimum distance for binary Reed–Muller codes. Later, in [Theorem 9.15](#), we will calculate the minimum distance for non-binary Reed–Muller codes.

Theorem 9.3

The minimum distance of $R(r, m)$ is 2^{m-r} .

Proof

By induction on m . If $m = r$, then the evaluation of the polynomial $X_1 \cdots X_r$ is a codeword of weight one.

Suppose that the minimum distance of $R(r, m)$ is 2^{m-r} .

Order the vectors of \mathbb{F}_2^{m+1} so that the first 2^m vectors have $x_{m+1} = 0$.

A codeword $(u, u + v)$ of $R(r, m + 1)$ is the evaluation of a polynomial

$$f(X) + X_{m+1}g(X),$$

where $f(X)$ is a polynomial of degree at most r in m variables and $g(X)$ is a polynomial of degree at most $r - 1$ in m variables. Then $u \in R(r, m)$, since it is the evaluation of $f(X)$ and $v \in R(r - 1, m)$, since it is the evaluation of $g(X)$.

If $u = 0$, then the codeword is $(0, v)$ and by induction has non-zero weight at least $2^{m-(r-1)}$.

If $u + v = 0$, then $u = v \in R(r - 1, m)$ and so the codeword $(u, 0)$ has non-zero weight at least $2^{m-(r-1)}$.

If neither u nor $u + v$ is zero, then $(u, u + v)$ has weight at least $2 \cdot 2^{m-r} = 2^{m-r+1}$, since both u and $u + v$ are in $R(r, m)$.

Thus, the minimum weight of a non-zero codeword of $R(r, m + 1)$ is 2^{m-r+1} . By [Lemma 4.1](#), the minimum weight of a non-zero codeword of a linear code is equal to its minimum distance. \square

9.2 Decoding Reed–Muller Codes

The popularity of Reed–Muller codes in real-world applications is due in part to the fact that there are fast decoding algorithms, the most common of which is the focus of this section. Before we consider this decoding algorithm, we first prove a couple of lemmas which prove some properties of Boolean functions.

For each non-empty subset J of $\{1, \dots, m\}$, let

$$f_J(X) = \prod_{j \in J} X_j$$

and define

$$f_{\emptyset}(X) = 1.$$

Then

$$\{f_J(X) \mid J \subseteq \{1, \dots, m\}, |J| \leq r\}$$

is a basis for the space of polynomials in m variables of degree at most r whose evaluations define Boolean functions.

We will exploit the following lemma repeatedly.

Lemma 9.4 *Let J be a subset of $\{1, \dots, m\}$. Suppose*

$$g(X) = \sum_{L \subseteq \{1, \dots, m\}} a_L f_L(X),$$

for some $a_L \in \mathbb{F}_2$, where the sum is over all subsets L of size at most $m - |J|$.

Then

$$\sum_{x \in \mathbb{F}_2^m} f_J(x)g(x) = a_{\{1, \dots, m\} \setminus J}.$$

Proof

Let $K \subseteq \{1, \dots, m\}$.

If there is an $i \in \{1, \dots, m\} \setminus K$, then

$$\sum_{\{x \in \mathbb{F}_2^m \mid x_i = 0\}} f_K(x) = \sum_{\{x \in \mathbb{F}_2^m \mid x_i \neq 0\}} f_K(x).$$

This implies

$$\sum_{x \in \mathbb{F}_2^m} f_K(x) = 0, \tag{9.1}$$

unless $K = \{1, \dots, m\}$.

Then

$$\sum_{x \in \mathbb{F}_2^m} f_J(x)g(x) = \sum_{L \subseteq \{1, \dots, m\}} \sum_{x \in \mathbb{F}_2^m} a_L f_J(x) f_L(x),$$

where the first sum on the right-hand side is over all subsets L of size at most $m - |J|$.

This expression is equal to

$$\sum_{L \subseteq \{1, \dots, m\}} a_L \sum_{x \in \mathbb{F}_2^m} f_{J \cup L}(x) = a_{\{1, \dots, m\} \setminus J},$$

by (9.1). □

Theorem 9.5

The dual of the code $R(r, m)$ is the Reed–Muller code $R(m - r - 1, m)$.

Proof

A codeword u of $R(r, m)$ is the evaluation of a polynomial

$$g(X) = \sum_{K \subseteq \{1, \dots, m\}} a_K f_K(X),$$

for some $a_K \in \mathbb{F}_2$, where the sum is over all subsets of size at most r .

A codeword v of $R(m - r - 1, m)$ is the evaluation of

$$h(X) = \sum_{L \subseteq \{1, \dots, m\}} b_L f_L(X),$$

for some $b_L \in \mathbb{F}_2$, where the sum is over all subsets of size at most $m - r - 1$.

The inner product of u and v is

$$\begin{aligned} \sum_{x \in \mathbb{F}_2^m} h(x)g(x) &= \sum_{x \in \mathbb{F}_2^m} \sum_K \sum_L a_K b_L f_K(x) f_L(x) \\ &= \sum_K \sum_L a_K b_L \sum_{x \in \mathbb{F}_2^m} f_{K \cup L}(x) = 0, \end{aligned}$$

by [Lemma 9.4](#).

Therefore,

$$R(m - r - 1, m) \subseteq R(r, m)^\perp.$$

By [Theorem 9.3](#), the sum of the dimensions of $R(r, m)$ and $R(m - r - 1, m)$ is 2^m , which is the length of the codes.

Hence,

$$\dim R(m - r - 1, m) = \dim R(r, m)^\perp.$$

□

The following lemma is fundamental to the decoding algorithm.

Lemma 9.6 *Let*

$$g(X) = \sum_{K \subseteq \{1, \dots, m\}, |K| \leq r} b_K f_K(X),$$

where $b_K \in \mathbb{F}_2$ and let J be a subset of $\{1, \dots, m\}$ of size r .
 For all 2^{m-r} choices of $a_i \in \mathbb{F}_2, i \in \{1, \dots, m\} \setminus J$,

$$\sum_{x \in \mathbb{F}_2^m} g(x) \prod_{i \in \{1, \dots, m\} \setminus J} (x_i + a_i) = b_J.$$

Proof

When we expand the product in the sum, all terms have degree less than m except those coming from

$$g(x) \prod_{i \in \{1, \dots, m\} \setminus J} x_i = g(x) f_{\{1, \dots, m\} \setminus J}(x).$$

The lemma follows from [Lemma 9.4](#). □

We are now in a position to describe a decoding algorithm for Reed–Muller codes, which is an example of a **majority-logic decoding** algorithm. Let v be the received vector, whose coordinates v_x are indexed by the vectors $x \in \mathbb{F}_2^m$. For each subset J of $\{1, \dots, m\}$ of size r , we perform a test. We wish to determine whether u_J is zero or one, where the sent codeword u is the evaluation of

$$\sum_{J \subseteq \{1, \dots, m\}, |J| \leq r} u_J f_J(X).$$

For all 2^{m-r} choices of $a_i \in \mathbb{F}_2, i \in \{1, \dots, m\} \setminus J$, we calculate

$$\sum_{x \in \mathbb{F}_2^m} v_x \prod_{i \in \{1, \dots, m\} \setminus J} (x_i + a_i).$$

If the result of this test is 1 in the majority of cases, then we conclude that $u_J = 1$ and vice versa, if it is 0 in the majority of cases, then we conclude that $u_J = 0$. Once we have completed this for all subsets J of $\{1, \dots, m\}$ of size r , we subtract the evaluation of

$$\sum_{K \subseteq \{1, \dots, m\}, |K|=r} u_K f_K(X),$$

from the received vector and continue with the subsets of size $r - 1$ supposing that, if we are correctly decoding, we now have a corrupted codeword of $R(r - 1, m)$.

All that remains to be shown, to prove that this decoding algorithm will correct up to $2^{m-r-1} - 1$ error bits, is to show that an error bit will only affect one of the tests. Before we prove this in [Lemma 9.8](#), we consider an example.

Example 9.7

Suppose that we have encoded using $R(2, 4)$ and have received

$$v = (1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0),$$

where the vectors of \mathbb{F}_2^4 are ordered as in the matrix G in [Example 9.1](#).

We calculate

$$w = vG^t = (1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0).$$

The coordinates are indexed by subsets of $\{1, 2, 3, 4\}$ of size at most 2, as in [Example 9.1](#).

Indexing the coordinates explicitly

$$\begin{array}{c|cccccccccccc} J & \emptyset & \{1\} & \{2\} & \{3\} & \{4\} & \{12\} & \{13\} & \{14\} & \{23\} & \{24\} & \{34\} \\ \hline w_J & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}$$

where

$$w_J = \sum_{x \in \mathbb{F}_2^m} v_x f_J(x)$$

are the coordinates of w .

We start by determining u_J for the subsets J of size $r = 2$.

To determine $u_{\{12\}}$, we make $2^{m-r} = 4$ tests by calculating

$$\sum_{x \in \mathbb{F}_2^m} v_x x_3 x_4, \quad \sum_{x \in \mathbb{F}_2^m} v_x (x_3 x_4 + x_3), \quad \sum_{x \in \mathbb{F}_2^m} v_x (x_3 x_4 + x_4)$$

and

$$\sum_{x \in \mathbb{F}_2^m} v_x (x_3 x_4 + x_3 + x_4 + 1),$$

which is

$$w_{\{34\}}, w_{\{34\}} + w_{\{3\}}, w_{\{34\}} + w_{\{4\}} \text{ and } w_{\{34\}} + w_{\{3\}} + w_{\{4\}} + w_{\emptyset},$$

respectively.

The results of these tests are 0, 1, 0, 0, respectively, so we decode $u_{\{12\}}$ as 0, since there are a majority of zeros.

The following table lists the results of these tests for all subsets of size 2 and indicates the majority decision.

$u_{\{12\}}$	0, 1, 0, 0 \rightarrow 0	$u_{\{13\}}$	1, 0, 1, 1 \rightarrow 1	$u_{\{14\}}$	0, 1, 1, 1 \rightarrow 1
$u_{\{23\}}$	0, 0, 0, 1 \rightarrow 0	$u_{\{24\}}$	1, 1, 0, 1 \rightarrow 1	$u_{\{34\}}$	1, 1, 0, 1 \rightarrow 1

Based on the results of those tests, we subtract

$$(0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1)G$$

from v and get

$$v^1 = v + (0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1)G = (1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0).$$

If we are decoding correctly, v^1 should be a (possibly) corrupted codeword of $R(1, 4)$. To determine u_J , where J is a subset of size 1, we repeat the above.

We calculate

$$w^1 = v^1 G_1^t,$$

where G_1 is the generator matrix of $R(3, m)$. This vector will have coordinates

$$w_K^1 = \sum_{x \in \mathbb{F}_2^m} f_K(x) v_x^1,$$

where K is a subset of $\{1, 2, 3, 4\}$ of size at most 3.

Indexing the coordinates explicitly as before

K	\emptyset	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{12\}$	$\{13\}$	$\{14\}$
w_K^1	1	0	1	1	0	0	0	0

K	$\{23\}$	$\{24\}$	$\{34\}$	$\{123\}$	$\{124\}$	$\{134\}$	$\{234\}$
w_K^1	1	0	0	0	0	1	0

allows us to perform $2^{m-(r-1)} = 8$ tests for each u_J .

To determine $u_{\{1\}}$, we make 8 tests by calculating

$$w_{\{234\}}^1, w_{\{234\}}^1 + w_{\{23\}}^1, w_{\{234\}}^1 + w_{\{24\}}^1, w_{\{234\}}^1 + w_{\{34\}}^1, w_{\{234\}}^1 + w_{\{23\}}^1 + w_{\{24\}}^1 + w_{\{2\}}^1,$$

$$w_{\{234\}}^1 + w_{\{23\}}^1 + w_{\{34\}}^1 + w_{\{3\}}^1, w_{\{234\}}^1 + w_{\{24\}}^1 + w_{\{34\}}^1 + w_{\{4\}}^1$$

and

$$w_{\{234\}}^1 + w_{\{23\}}^1 + w_{\{24\}}^1 + w_{\{34\}}^1 + w_{\{2\}}^1 + w_{\{3\}}^1 + w_{\{4\}}^1 + w_{\emptyset}^1.$$

The results of these tests are

$u_{\{1\}}$	$0, 1, 0, 0, 0, 0, 0, 0 \rightarrow 0$	$u_{\{2\}}$	$1, 1, 1, 1, 1, 0, 1, 1 \rightarrow 1$
$u_{\{3\}}$	$0, 0, 0, 0, 0, 1, 0, 0 \rightarrow 0$	$u_{\{4\}}$	$0, 0, 0, 1, 0, 0, 0, 0 \rightarrow 0$

Based on the results of the tests, we subtract

$$(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)G$$

from v_1 and get

$$v_2 = v + (0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1)G = (1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1).$$

Summing

$$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)G$$

to v_2 we have that

$$v + (1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1)G = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0).$$

Therefore, we have determined that the error is in the 7-th bit, that the uncoded string

$$u = (1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1)$$

and that the sent codeword was uG . ■

To finish this section, we prove that an error bit affects exactly one of the tests when testing a corrupted codeword of $R(r, m)$. Since we perform 2^{m-r} tests, this implies that we can correct up to $2^{m-r-1} - 1$ error bits, or in other terms $\frac{1}{2}d - 1$ error bits, since the minimum distance d of $R(r, m)$ is 2^{m-r} .

Lemma 9.8 *Suppose that e is a vector of $\mathbb{F}_2^{2^m}$ of weight one, whose coordinates are indexed by the vectors of \mathbb{F}_2^m . Let J be a subset of $\{1, \dots, m\}$. For all but one of the choices of a , whose coordinates $a_i \in \mathbb{F}_2$ for $i \in \{1, \dots, m\} \setminus J$,*

$$\sum_{x \in \mathbb{F}_2^m} e_x \prod_{i \in \{1, \dots, m\} \setminus J} (x_i + a_i) = 0,$$

where e_x is the coordinate of e indexed by x .

Proof

Let y be the vector of \mathbb{F}_2^m indexing the coordinate where the vector e has a 1.

The vector e is the evaluation of

$$\prod_{i=1}^m (X_i + y_i + 1),$$

since it is zero unless $X_i = y_i$ for all $i = 1, \dots, m$.

Hence, for all $x \in \mathbb{F}_2^m$,

$$e_x \prod_{i \in \{1, \dots, m\} \setminus J} (x_i + a_i) = \prod_{i=1}^m (x_i + y_i + 1) \prod_{i \in \{1, \dots, m\} \setminus J} (x_i + a_i),$$

which will contain a factor $x_i^2 + x_i$ (and is therefore zero) unless $a_i = y_i + 1$ for all $i \in \{1, \dots, m\} \setminus J$. \square

To decode using this majority-logic decoding algorithm we perform at most 2^m tests k times, where k is the dimension of the code. This is less than n^2 tests, where n is the length of the code. Each test involves summing less than n terms, so the decoding algorithm is completed in a number of steps which is polynomial in the length of the code. This should be compared to syndrome decoding from [Chapter 4](#), which involved searching through a look-up table with a number of entries which is exponential in n . For this reason Reed–Muller codes and the majority-logic decoding algorithm are widely implemented. However, they do not give a sequence of asymptotically good codes. Although the relative minimum distance is 2^{-r} , which we can bound away from zero by fixing r , the transmission rate of $R(r, m)$ is less than

$$\frac{r}{n} \binom{\log n}{r}$$

which tends to zero as n tends to infinity.

9.3 Kerdock Codes

A codeword of $R(2, m) \setminus R(1, m)$ is the evaluation of polynomials of the form

$$q(X) + \ell(X) \text{ or } q(X) + \ell(X) + 1,$$

where $\ell(X)$ is a linear form in m variables and

$$q(X) = \sum_{1 \leq i < j \leq m} a_{ij} X_i X_j$$

is a non-zero quadratic form.

If the quadratic form $q(X)$ has maximum rank, then we will prove that, for all the linear forms $\ell(X)$, these codewords will have large weight. Therefore, if we can find a set of quadratic forms whose differences are quadratic forms of maximum rank, then the distance between any two codewords will be large. In this section we will develop and formalise this idea.

Let $A = (a_{ij})$ be the symmetric matrix defined by the symmetric bilinear form

$$b(X, Y) = q(X + Y) - q(X) - q(Y) = \sum_{1 \leq i < j \leq m} a_{ij} (X_i Y_j + X_j Y_i) = X^t A Y.$$

The **rank** of the bilinear form $b(X, Y)$ is defined to be the rank of A .

Lemma 9.9 *Suppose m is even. The evaluation of*

$$\sum_{i=1}^{m/2} X_{2i-1} X_{2i}$$

at the vectors of \mathbb{F}_2^m has $2^{m-1} + 2^{m/2-1}$ zeros.

Proof

There are $2^{m/2}$ zeros of the form $(0, x_2, 0, x_4, \dots, 0, x_m)$.

If $x_{2i-1} \neq 0$ for some $i = 1, \dots, m/2$, then one of the x_{2i} is determined by

$$\sum_{i=1}^{m/2} x_{2i-1} X_{2i} = 0,$$

which gives $2^{m/2-1}(2^{m/2} - 1)$ zeros of this form, $2^{m/2-1}$ zeros for each non-zero vector $(x_1, x_3, \dots, x_{m-1})$.

Hence, there are precisely $2^{m-1} + 2^{m/2-1}$ zeros when evaluated at the vectors of \mathbb{F}_2^m . \square

We are going to construct codes whose codewords are the evaluation of the sum of a quadratic form and a linear form. For this reason, we want to know the weights of the vectors which are the evaluations of these Boolean functions.

Lemma 9.10 *Suppose m is even, $q(X)$ is a quadratic form and $\ell(X)$ is a linear form. If the bilinear form associated to $q(X)$ has rank m , then the evaluation of $q(X) + \ell(X)$ at the vectors of \mathbb{F}_2^m has either $2^{m-1} + 2^{m/2-1}$ or $2^{m-1} - 2^{m/2-1}$ zeros.*

Proof

Dickson's theorem, [Exercise 9.2](#), implies that there is a basis of \mathbb{F}_2^m with respect to which $q(x) + \ell(X)$ is

$$\sum_{i=1}^{m/2} (X_{2i-1} X_{2i} + a_{2i-1} X_{2i-1} + a_{2i} X_{2i}).$$

This is equal to

$$\sum_{i=1}^{m/2} (X_{2i-1} + a_{2i})(X_{2i} + a_{2i-1}) + b$$

for some $b \in \mathbb{F}_2$. By [Lemma 9.9](#), the evaluation of $q(X) + \ell(X)$ has either $2^{m-1} + 2^{m/2-1}$ zeros or $2^m - (2^{m-1} + 2^{m/2-1})$ zeros, depending on whether $b = 0$ or 1 . \square

Let K be a set of symmetric $m \times m$ matrices over \mathbb{F}_2 , which have zeros on the diagonal, and which have the property that the matrix $A - A'$ has rank m for all distinct $A, A' \in K$.

No two matrices in K can have the same first row, since their difference is of rank m . The entries on the diagonal of the matrices in K are zero, so the top-left entry of a matrix in K is zero. Hence, we have that

$$|K| \leq 2^{m-1}.$$

For each $A = (a_{ij}) \in K$, let

$$q_A(X) = \sum_{1 \leq i < j \leq m} a_{ij} X_i X_j.$$

Let $C(K)$ be the code whose codewords are the evaluation at the vectors of \mathbb{F}_2^m of

$$q_A(X) + \ell(X) \text{ or } q_A(X) + \ell(X) + 1,$$

for all $A \in K$ and for all linear forms $\ell(X)$.

Theorem 9.11

Suppose that m is even. The code $C(K)$ is a binary block code of length 2^m , size $|K||R(1, m)|$ and minimum distance $2^{m-1} - 2^{m/2-1}$.

Proof

The distance between the evaluation of

$$q_A(X) + \ell(X) + b$$

and

$$q_{A'}(X) + \ell'(X) + b'$$

is the weight of the evaluation of

$$q_{A-A'}(X) + \ell(X) - \ell'(X) + b - b'.$$

Since, $A - A'$ has rank m , Lemma 9.10 implies that this distance is at least $2^{m-1} - 2^{m/2-1}$.

□

A **Kerdock code** is a code $C(K)$ where $|K| = 2^{m-1}$. Thus, for a Kerdock code, K is of maximum size and the set K is called a **Kerdock set**. A Kerdock code is a binary

block code of length 2^m , it has minimum distance $2^{m-1} - 2^{m/2-1}$ and size 2^{2m} , i.e. it is a $(2^m, 2^{2m}, 2^{m-1} - 2^{m/2-1})_2$ code.

There are many non-equivalent Kerdock codes. Indeed, if $m - 1$ is not prime, then there are at least $2^{\sqrt{m}/2}$ inequivalent Kerdock codes of length 2^m . However, a sequence of Kerdock codes, whose lengths tend to infinity, is asymptotically bad. Although the relative minimum distance tends to $\frac{1}{2}$, the transmission rate is $2m/2^m$, which tends to zero.

Kerdock codes are of interest because they can be non-linear. The algebraic and geometric nature of their construction allows for non-trivial decoding algorithms to be implemented. The fact that Kerdock codes can be non-linear opens up the possibility of constructing codes with parameter sets for which linear codes do not exist.

Example 9.12

Consider the set of 4×4 matrices over \mathbb{F}_2

$$\left\{ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \right\}.$$

This set of matrices can be extended to a set K of 8 matrices with the property that the difference of any two matrices has rank 4, see [Exercise 9.5](#). Thus, K is a Kerdock set and, by [Theorem 9.11](#), the binary Kerdock code $C(K)$ is a $(16, 256, 6)_2$ code. We proved in [Example 4.20](#) that there is no binary linear code with these parameters. This code is the **Nordstrom–Robinson** code. ■

9.4 Non-binary Reed–Muller Codes

Until now we have only considered Reed–Muller codes over \mathbb{F}_2 , but one can naturally generalise the definition of a Reed–Muller code over a general finite field \mathbb{F}_q . The codewords of $R_q(r, m)$ are the evaluations at the vectors of \mathbb{F}_q^m of polynomials of degree at most r in m variables, where the degree in any particular variable is at most $q - 1$. The number of vectors in an m -dimensional vector space over \mathbb{F}_q is q^m , so the length of the linear code $R_q(r, m)$ is q^m . Its dimension is more difficult to calculate, see [Exercise 9.7](#). In the following examples, we calculate the dimension for some specific cases and some low weight codewords, which we will then go on and prove are of minimum non-zero weight.

Example 9.13

Suppose $r \leq q - 1$. The evaluation of any polynomial in m variables of degree at most r will be a codeword of $R_q(r, m)$. The set

$$\{X_1^{c_1} \cdots X_m^{c_m} \mid c_1 + \cdots + c_m \leq r\}$$

is a basis for the space of polynomials in m variables of degree at most r . Hence, the dimension of $R_q(r, m)$ is

$$\binom{m+r}{r}$$

since this is the number of non-negative integer solutions to

$$c_1 + \cdots + c_m \leq r.$$

Let $g(X_1)$ be a polynomial of degree r with r distinct roots in \mathbb{F}_q . The evaluation of g is a codeword with precisely rq^{m-1} zero coordinates; a zero coordinate being indexed by a vector of \mathbb{F}_q^m whose first coordinate is a root of g . Therefore, $R_q(r, m)$ has codewords of weight $(q-r)q^{m-1}$. ■

Example 9.14

The space of polynomials of degree at most 3 in three variables in which no variable has an exponent larger than 2 has a basis

$$\{1, X_1, X_2, X_3, X_1^2, X_2^2, X_3^2, X_1X_2, X_1X_3, X_2X_3, X_1^2X_2, X_1^2X_3, X_2^2X_1, X_2^2X_3, X_3^2X_1, X_3^2X_2, X_1X_2X_3\}.$$

Therefore, the code $R_3(3, 3)$ is a 17-dimensional ternary linear code of length 27. We could also have arrived at this conclusion by considering a monomial basis for all polynomials of degree at most three in three variables and deleting X_1^3, X_2^3 and X_3^3 , see [Exercise 9.7](#). ■

Suppose that $r = a(q - 1) + b$, where $0 \leq b \leq q - 2$. If $g(X_1)$ is a polynomial of degree b with b distinct roots in \mathbb{F}_q , then the evaluation of

$$g(X_1)(X_2^{q-1} - 1) \cdots (X_{a+1}^{q-1} - 1),$$

a polynomial of degree r , is non-zero only when evaluated at

$$x = (x_1, 0, \dots, 0, x_{a+2}, \dots, x_m)$$

for some x_1 which is not a root of g . Therefore, $R_q(r, m)$ has a codeword of weight

$$(q - b)q^{m-a-1}.$$

We shall prove that this is the minimum weight of a non-zero codeword in the following theorem, the proof of which is an example of a proof using the polynomial method. This type of proof, which one sees often in combinatorics, attempts to obtain bounds from the fact that the number of zeros of a non-zero polynomial is bounded. The application of the method is often something like the following. Given



a combinatorial object, a polynomial is constructed in such a way that the properties of the combinatorial object are translated into algebraic properties of the polynomial. Usually we are interested in the zeros of the polynomial, often restricted to subsets of a vector space. Here, the polynomial is directly given as the polynomial whose evaluation is the codeword. By bounding from above the number of zeros of the polynomial, we will bound from below the weight of the codeword.

Theorem 9.15

The minimum distance of $R_q(r, m)$ is $(q - b)q^{m-a-1}$, where $r = a(q - 1) + b$ and $0 \leq b \leq q - 2$.

Proof

By induction on m .

If $m = 1$, then the codewords are the evaluation of a polynomial of degree $r \leq q - 1$ in one variable. The polynomial has at most r zeros, so the codeword has weight at least $q - r$. Observe that if $r = q - 1$, then $a = 1$ and $b = 0$ and

$$(q - b)q^{m-a-1} = q/q = 1 = q - r.$$

Suppose that the codeword $u \in R_q(r, m)$ is the evaluation of the polynomial

$$f(X) = f(X_1, \dots, X_m).$$

We write $f(X)$ as a polynomial in X_m , whose coefficients are polynomials in X_1, \dots, X_{m-1} . Thus,

$$f(X) = \sum_{i=0}^c f_i(X_1, \dots, X_{m-1})X_m^i,$$

where c is the degree of $f(X)$ in the indeterminate X_m . Note that $f_c(X_1, \dots, X_{m-1}) \neq 0$ and

$$\deg f_c \leq \deg f - c \leq r - c.$$

The codeword of $R_q(r - c, m - 1)$ which is the evaluation of f_c has, by induction, at least

$$(q - b')q^{m-a'-2}$$

non-zero coordinates, where $r - c = a'(q - 1) + b'$ and $0 \leq b' \leq q - 2$.

For any (x_1, \dots, x_{m-1}) such that $f_c(x_1, \dots, x_{m-1}) \neq 0$, there are at least $q - c$ elements of \mathbb{F}_q for which $f(x_1, \dots, x_{m-1}, X_m)$ is not zero. Hence, the codeword u has weight at least

$$(q - c)(q - b')q^{m-a'-2}.$$

It remains to prove that

$$(q - c)(q - b')q^{m-a'-2} \geq (q - b)q^{m-a-1}.$$

The theorem then follows since, by [Lemma 4.1](#), the minimum distance of a linear code is equal to the minimum weight of a non-zero codeword.

If $a' \leq a - 2$, then this is clear, so we can assume $a' = a - 1$ or a .

Suppose $a' = a - 1$ and $(q - c)(q - b') < q - b$. This inequality implies $b' > b$. We have $r = a(q - 1) + b$ and $r - c = a'(q - 1) + b'$, so

$$c = (a - a')(q - 1) + b - b' = q - 1 + b - b'.$$

Then $(q - c)(q - b') < (q - b)$ implies $(b' - b + 1)(q - b') < q - b$, a contradiction.

Suppose $a' = a$ and $(q - c)(q - b') < q(q - b)$. We have $r = a(q - 1) + b$ and $r - c = a'(q - 1) + b'$, so $c = b - b'$. Then $(q - c)(q - b') < q(q - b)$ implies $(q - b + b')(q - b') < q(q - b)$ which implies $b < b'$ and $c < 0$, a contradiction. □

9.5 Comments

Reed–Muller codes were introduced by Reed [\[59\]](#) and Muller [\[53\]](#) in the 1950s.

We have taken an algebraic rather than a geometric approach to the majority-logic decoding algorithm. For a geometric description of the algorithm, see Van Lint [\[74\]](#) or MacWilliams and Sloane [\[50\]](#).

Dickson's classification of quadratic form over fields of even characteristic is from [\[22\]](#).

If m is odd, then there are examples of sets K for which [Exercise 9.6](#) is a $(\frac{1}{2}(m^2 + m) + 1 - rm)$ -dimensional binary linear code. The 11-dimensional codes ($m = 5$ and $r = 2$) are the codes which caused a dispute between Apple and Samsung, referred to in James Davis' lecture [\[20\]](#). They can be found in Corollary 17 ($m = 5$, $d = t = 2$) on page 455 of MacWilliams and Sloane [\[50\]](#).

Kerdock codes were first considered by Kerdock in [\[44\]](#) in 1972. That there are an exponential number of inequivalent Kerdock codes is proven by Kantor in [\[41\]](#). Kantor takes a geometric approach to Kerdock codes in the articles [\[42\]](#), a treatment of which can be found in Chapter 12 of Cameron and van Lint's book [\[17\]](#). The Nordstrom–Robinson code is from [\[54\]](#). Kerdock codes have applications to quantum mechanics, see [\[15\]](#) and [\[18\]](#).

The non-binary Reed–Muller codes were defined by various authors. [Theorem 9.15](#) is attributed to Kasami, Lin and Peterson [\[43\]](#) in Bishnoi [\[11\]](#), where the proof given here is adapted from.

9.6 Exercises

9.1 Suppose that we have sent a codeword of the code $R(2, 4)$, the coordinates ordered as in [Example 9.1](#), and have received the vector

$$(0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1).$$

- i. Decode the received vector using syndrome decoding.
- ii. Decode the received vector using majority-logic decoding.

9.2 Suppose that $q(X)$ is a quadratic form of rank m of the type

$$q(X) = \sum_{1 \leq i < j \leq m} q_{ij} X_i X_j.$$

Prove that there is a basis of \mathbb{F}_2^m , with respect to which, $q(X)$ is

$$\sum_{i=1}^{m/2} X_{2i-1} X_{2i}.$$

9.3

- i. Prove that we can select half the codewords of $R(1, m)$ so that the $2^m \times 2^m$ matrix H , whose rows are the selected codewords with zeros changed to minus one, has the property that $HH^t = 2^m I$, where I is the $2^m \times 2^m$ identity matrix.
- ii. Prove that for each vector $v \in \mathbb{F}_2^{2^m}$ there is a codeword u of $R(1, m)$ such that $d(u, v) \leq 2^{m-1} - 2^{m/2-1}$.

9.4 Prove that the Kerdock code $C(K)$ of length 2^m is linear if and only if the Kerdock set K is a subspace of the vector space of $m \times m$ matrices.

9.5 Complete the set of matrices to a Kerdock set K of eight matrices and prove that $C(K)$ is a non-linear $(16, 256, 16)$ code.

$$\left\{ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \right\}$$

9.6

- i. Prove that the evaluation of

$$q(X) = \sum_{i=1}^r X_{2i-1} X_{2i}$$

at the vectors of \mathbb{F}_2^m , has $2^{m-1} + 2^{m-r-1}$ zeros.

- ii. Prove that the evaluation of $q(X) + \ell(X)$, where $\ell(X)$ is a linear form and $q(X)$ is a quadratic form whose associated bilinear form is of rank $2r$, at the vectors of \mathbb{F}_2^m , has either $2^{m-1} + 2^{m-r-1}$, 2^{m-1} or $2^{m-1} - 2^{m-r-1}$ zeros.
- iii. Suppose that K is a set of $m \times m$ symmetric matrices over \mathbb{F}_2 with the property that $A + A'$ has rank at least $2r$ for all $A, A' \in K$. Construct a $(2^m, |K|2^{m+1}, 2^{m-1} - 2^{m-r-1})_2$ code.
- iv. Use iii. to construct a $[32, 11, 12]_2$ code.
- v. Construct a linear code with the same parameters from the code of length 31 constructed in Exercise 5.6.

9.7

- i. By finding a monomial basis for the space of polynomials in 3 variables of degree at most 4, in which the degree of each variable is at most 2, calculate the dimension of $R_3(4, 3)$.
- ii. Prove that if $r \leq q - 1$, then the dimension of $R_q(r, m)$ is

$$\sum_{i=0}^r \binom{m+i-1}{m-1}.$$

- iii. Prove that the dimension of $R_q(r, m)$ is

$$\sum_{i=0}^r \sum_{j=0}^m (-1)^j \binom{m+i-1-jq}{m-1} \binom{m}{j}.$$



p -Adic Codes

The p -adic numbers were first considered by Hensel in the 19th century. He observed that the primes play an analogous role in the integers as linear polynomials do in $\mathbb{C}[X]$. The Laurent expansion of a rational function led him to consider the p -adic expansion of a rational number. In this chapter, for a fixed prime p , we will construct block codes over the rings $\mathbb{Z}/p^h\mathbb{Z}$ simultaneously, by constructing codes over the p -adic numbers and then considering the coordinates modulo p^h . These codes will be linear over the ring but when mapped to codes over $\mathbb{Z}/p\mathbb{Z}$ will result in codes which are not equivalent to linear codes. We start with a brief introduction to p -adic numbers, which will cover enough background for our purposes. The classical cyclic codes, that we constructed in [Chapter 5](#), lift to cyclic codes over the p -adic numbers. In the case of the cyclic Hamming code, this lift extends to a code over $\mathbb{Z}/4\mathbb{Z}$ which, when mapped to a binary code, gives a non-linear code with a set of parameters for which no linear code exists.

10.1 p -Adic Numbers

Let p be a prime.

The set of p -**adic integers**, which is denoted by \mathbb{Z}_p , is the set of sequences,

$$a = (a_1, a_2, a_3, \dots),$$

where $a_i \in \mathbb{Z}/p^i\mathbb{Z}$ for all $i \in \mathbb{N}$ and

$$a_{j+1} \equiv a_j \pmod{p^j}.$$

An ordinary integer $n \in \mathbb{Z}$ is an element of \mathbb{Z}_p defined by the sequence

$$a_j \equiv n \pmod{p^j}.$$

The sequence defined by

$$a_{j+1} = a_j + p^j,$$

$j \in \mathbb{N}$, is a p -adic integer which is not an ordinary integer.

For example, with $a_1 = 3$ and $p = 5$, this sequence begins

$$(3, 8, 33, 158, 783, \dots).$$

We define addition and multiplication on the sequences component-wise, so

$$a + b = (a_1, a_2, a_3, \dots) + (b_1, b_2, b_3, \dots) = (a_1 + b_1, a_2 + b_2, a_3 + b_3, \dots).$$

To verify $a + b \in \mathbb{Z}_p$, observe that

$$a_{j+1} + b_{j+1} \equiv a_j + b_j \pmod{p^j}.$$

Similarly,

$$ab = (a_1, a_2, a_3, \dots)(b_1, b_2, b_3, \dots) = (a_1b_1, a_2b_2, a_3b_3, \dots).$$

To verify $ab \in \mathbb{Z}_p$, observe that

$$a_{j+1}b_{j+1} \equiv a_jb_j \pmod{p^j}.$$

With these definitions multiplication is distributive with respect to addition, so \mathbb{Z}_p is a ring and has a multiplicative identity element

$$1 = (1, 1, 1, \dots).$$

If a is a sequence for which $a_1 = 0$, then a does not have a multiplicative inverse, so \mathbb{Z}_p is not a field. It is, however, an integral domain ($xy = 0$ implies either $x = 0$ or $y = 0$), so it has a quotient field. This quotient field is called the **field of p -adic numbers** and is denoted \mathbb{Q}_p . Elements of \mathbb{Q}_p are called **p -adic numbers**.

All non-zero elements of \mathbb{Z}_p can be written as the product of a unit times some non-negative power of p .

For example, the 5-adic integer

$$(0, 15, 40, 290, 915, \dots) = 5(3, 8, 58, 183, \dots),$$

since $15 \equiv 0$ modulo 5, $40 \equiv 15$ modulo 25, $290 \equiv 40$ modulo 125, etc.

The field \mathbb{Q}_p consists of the sequences where we allow negative powers of p as well.

For example,

$$5^{-2}(2, 22, 97, 222, \dots),$$

10.2 · Polynomials over the p -Adic Numbers

is a 5-adic number.

The product of $p^\alpha(a_1, a_2, a_3, \dots)$ and $p^\beta(b_1, b_2, b_3, \dots)$ is

$$p^{\alpha+\beta}(a_1b_1, a_2b_2, a_3b_3, \dots).$$

Returning to the previous examples,

$$5(3, 8, 58, 183, \dots)5^{-2}(2, 22, 97, 222, \dots) = 5^{-1}(1, 1, 1, 1, \dots).$$

10.2 Polynomials over the p -Adic Numbers

Let $\overline{\mathbb{Q}}_p$ denote an algebraic closure of \mathbb{Q}_p . Recall that, since $\overline{\mathbb{Q}}_p$ is an algebraic closure, the polynomials of positive degree over \mathbb{Q}_p factorise into linear factors over $\overline{\mathbb{Q}}_p$. The following lemma is a straightforward application of the binomial theorem.

Lemma 10.1 *If $\alpha, \beta \in \overline{\mathbb{Q}}_p$ and*

$$\alpha \equiv \beta \pmod{p^r}$$

then

$$\alpha^p \equiv \beta^p \pmod{p^{r+1}}.$$

Proof

We can write $\alpha = \beta + p^r\gamma$, for some $\gamma \in \mathbb{Z}_p$. Then

$$\alpha^p = (\beta + p^r\gamma)^p \equiv \beta^p \pmod{p^{r+1}}.$$

□

In ► [Chapter 2](#) we studied how to factorise cyclotomic polynomials over finite fields and put this to use in ► [Chapter 5](#) while constructing cyclic codes. The following theorem tells us that a factorisation over \mathbb{F}_p “lifts” to a factorisation over the p -adic numbers. As in ► [Chapter 5](#), we will exploit this factorisation to construct cyclic codes and their extensions with some surprising results.

Theorem 10.2

Let p be a prime and let n be a positive integer which is not a multiple of p . If h is a monic irreducible divisor of $X^n - 1$ in $(\mathbb{Z}/p\mathbb{Z})[X]$, then there exists a monic irreducible polynomial h_∞ in $\mathbb{Z}_p[X]$ which divides $X^n - 1$ and is congruent to h modulo p .

Proof

By induction on r , we will find a polynomial $h_r(X) \in (\mathbb{Z}/p^r\mathbb{Z})[X]$ such that $h_r(X)$ divides $X^n - 1$ and $h_r \equiv h$ modulo p . Then h_∞ will be the polynomial h_r as $r \rightarrow \infty$.

An element $c \in \mathbb{Z}/p^r\mathbb{Z}$ can be extended to an element of \mathbb{Z}_p by taking the sequence

$$(c_1, c_2, \dots, c_{r-1}, c, c, c, \dots),$$

where $c_i = c \bmod p^i$, for $i = 1, \dots, r-1$. Therefore, the coefficients of $h_r(X)$ can be viewed as elements of \mathbb{Z}_p and therefore as elements of $\overline{\mathbb{Q}_p}$.

Since n is not a multiple of p , the roots of $h_1(X)$ in $\overline{\mathbb{Q}_p}$ are distinct. By induction, we can assume that the roots of $h_r(X)$ are distinct.

For each root α of $h_r(X)$ (in $\overline{\mathbb{Q}_p}$),

$$\alpha^n \equiv 1 \pmod{p^r}.$$

Let

$$f(X) = h_r(X) + p^r g(X),$$

for some polynomial $g(X) \in \mathbb{Z}_p[X]$.

For each root β of f , there is a root α of $h_r(X)$ such that

$$\beta \equiv \alpha \pmod{p^r}.$$

Then, by [Lemma 10.1](#),

$$\beta^p \equiv \alpha^p \pmod{p^{r+1}}.$$

[Lemma 10.1](#) also implies that

$$\alpha^{np} \equiv 1 \pmod{p^{r+1}}$$

from which we deduce that

$$\beta^{np} \equiv 1 \pmod{p^{r+1}}.$$

Let

$$h_{r+1}(X) = \prod (X - \beta^p),$$

where the product runs over the roots β of f .

Then h_{r+1} divides $X^n - 1$ modulo p^{r+1} . Since

$$\beta^p \equiv \alpha^p \equiv \alpha \pmod{p},$$

h_{r+1} and h_r have the same roots modulo p .

Thus, the roots of h_{r+1} are distinct and

$$h_{r+1} \equiv h_r \pmod{p}.$$

□

10.3 p -Adic Codes

Let R be a commutative ring with multiplicative identity 1. An R -**module** M is a commutative group with a left multiplication from $R \times M \rightarrow M$ satisfying $\lambda(u + v) = \lambda u + \lambda v$, $(\lambda + \mu)u = \lambda u + \mu u$, $(\lambda\mu)u = \lambda(\mu u)$ and $1u = u$, for all $u, v \in M$ and all $\lambda, \mu \in R$.

The set \mathbb{Z}_p^n of n -tuples over the p -adic integers is a commutative group with respect to addition. We define left multiplication of an element $(u_1, \dots, u_n) \in \mathbb{Z}_p^n$ by an element $\lambda \in \mathbb{Z}_p$ as

$$\lambda(u_1, \dots, u_n) = (\lambda u_1, \dots, \lambda u_n).$$

This scalar multiplication satisfies $\lambda(u + v) = \lambda u + \lambda v$, $(\lambda + \mu)u = \lambda u + \mu u$, $(\lambda\mu)u = \lambda(\mu u)$ and $1u = u$, for all $u, v \in \mathbb{Z}_p^n$ and all $\lambda, \mu \in \mathbb{Z}_p$. Thus, with this scalar multiplication \mathbb{Z}_p^n is a \mathbb{Z}_p -module.

A **submodule** C of \mathbb{Z}_p^n is a non-empty subset of \mathbb{Z}_p^n which is closed under linear combinations. In other words,

$$\lambda u + \mu v \in C,$$

for all $u, v \in C$ and all $\lambda, \mu \in \mathbb{Z}_p$.

We now re-define the analogous objects that we saw for linear codes over a field for codes over \mathbb{Z}_p . A **p -adic code** of length n is a subset of \mathbb{Z}_p^n . A **linear code** over \mathbb{Z}_p is a submodule of \mathbb{Z}_p^n .

A **generator matrix** for a linear code C over \mathbb{Z}_p is a $k \times n$ matrix G with the property that

$$C = \{(u_1, \dots, u_k)G \mid (u_1, \dots, u_k) \in \mathbb{Z}_p^k\}.$$

We define the **scalar product** on \mathbb{Z}_p^n as the standard inner product

$$u \cdot v = u_1 v_1 + \dots + u_n v_n.$$

The **dual code** of a linear code C is defined, as in the case of a linear code over a finite field, as

$$C^\perp = \{v \in \mathbb{Z}_p^n \mid u \cdot v = 0 \text{ for all } u \in C\}.$$

A linear code C is **cyclic** if

$$(c_1, c_2, \dots, c_n) \in C$$

implies

$$(c_n, c_1, \dots, c_{n-1}) \in C.$$

A codeword of the cyclic code corresponds to a polynomial in the ring $\mathbb{Z}_p[X]/(X^n - 1)$ under the correspondence

$$(c_1, c_2, \dots, c_n) \mapsto c_1 + c_2X + \dots + c_nX^{n-1}.$$

As in the case of finite fields, under this correspondence, a cyclic code is an ideal $\langle g \rangle$, where g is some divisor of $X^n - 1$.

Example 10.3

The polynomial $X^3 + X + 1$ divides $X^7 - 1$ in $(\mathbb{Z}/2\mathbb{Z})[X]$. [Theorem 10.2](#) implies the existence of a polynomial in $\mathbb{Z}_2[X]$ which divides $X^7 - 1$. One can verify that

$$g(X) = X^3 + \lambda X^2 + (\lambda - 1)X - 1$$

divides $X^7 - 1$ in $\mathbb{Z}_2[X]$ if and only if $\lambda^2 - \lambda + 2 = 0$ by observing that

$$X^7 - 1 = (X^3 + \lambda X^2 + (\lambda - 1)X - 1)(X^3 + (1 - \lambda)X^2 - \lambda X - 1)(X - 1).$$

To calculate λ , suppose

$$\lambda = (a_1, a_2, a_3, \dots).$$

Since $a_1 \in \mathbb{Z}/2\mathbb{Z}$, we have $a_1 = 0$ or 1 .

If $a_1 = 0$, then substituting $\lambda \equiv 0 + 2a_2 \pmod{4}$ in

$$\lambda^2 - \lambda + 2 \equiv 0 \pmod{4}$$

implies

$$-2a_2 + 2 \equiv 0 \pmod{4},$$

so $a_2 = 1$ and $\lambda \equiv 2 \pmod{4}$.

Substituting $\lambda \equiv 2 + 4a_3 \pmod{8}$ in

$$\lambda^2 - \lambda + 2 \equiv 0 \pmod{8}$$

10.4 · Codes over $\mathbb{Z}/p^h\mathbb{Z}$

implies

$$4 - 2 - 4a_3 + 2 \equiv 0 \pmod{8},$$

so $a_3 = 1$ and $\lambda \equiv 6 \pmod{8}$.

Continuing in this way we deduce that one of the roots of $\lambda^2 - \lambda + 2$ is

$$\lambda = (0, 2, 6, 6, 6, 38, 38, 166, 422, \dots).$$

The cyclic code $\langle g \rangle$ is a 2-adic linear code of length 7 with generator matrix

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

■

To make use of these p -adic codes, we will now consider the coordinates of the codewords of a p -adic code modulo p^h for some h . The resulting code will be a code defined over the finite alphabet $\mathbb{Z}/p^h\mathbb{Z}$. We will use the matrix G from [Example 10.3](#) in [Example 10.9](#).

10.4 Codes over $\mathbb{Z}/p^h\mathbb{Z}$

A **linear code** over $\mathbb{Z}/p^h\mathbb{Z}$ is a $(\mathbb{Z}/p^h\mathbb{Z})$ -submodule of $(\mathbb{Z}/p^h\mathbb{Z})^n$. As in the case for a linear code over \mathbb{F}_q , we define a **generator matrix** for a linear code C over $(\mathbb{Z}/p^h\mathbb{Z})^n$ as a $r \times n$ matrix G with the property that

$$C = \{(u_1, \dots, u_r)G \mid (u_1, \dots, u_r) \in (\mathbb{Z}/p^h\mathbb{Z})^r\}.$$

If all the elements in the i -th row of G are divisible by p^j , then we can restrict u_i to $\mathbb{Z}/p^{h-j}\mathbb{Z}$.

Example 10.4

Let

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 5 & 6 & 6 & 7 & 7 & 8 & 8 & 8 \\ 0 & 0 & 3 & 6 & 3 & 6 & 3 & 6 & 3 & 6 & 3 & 6 & 3 & 6 & 3 & 6 & 3 & 6 & 3 & 6 & 3 \end{pmatrix},$$

where the elements of G are from $\mathbb{Z}/9\mathbb{Z}$.

The code generated by the matrix G is

$$C = \{(u_1, u_2, u_3)G \mid u_1, u_2 \in \mathbb{Z}/9\mathbb{Z}, u_3 \in \mathbb{Z}/3\mathbb{Z}\}.$$

Thus, the code C is a 9-ary code of length 20 of size 243.

The codeword

$$(3, 0, 1)G = (3, 0, 3, 6, 6, 0, 6, 0, 6, 0, 6, 0, 6, 0, 6, 0, 6, 0, 6, 0)$$

and the all-zero codeword differ in 11 coordinates, so the minimum distance is at most 11. It is [Exercise 10.3](#) to verify that the minimum distance is 11. ■

Theorem 10.5

After a suitable permutation of the coordinates, a linear code C over $(\mathbb{Z}/p^h\mathbb{Z})^n$ has a generator matrix of the form

$$G = \begin{pmatrix} I & A_{01} & A_{02} & A_{03} & \cdots & A_{0,h-1} & A_{0,h} \\ 0 & pI & pA_{12} & pA_{13} & \cdots & pA_{1,h-1} & pA_{1,h} \\ 0 & 0 & p^2I & p^2A_{23} & \cdots & p^2A_{2,h-1} & p^2A_{2,h} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdot & \cdots & 0 & 0 & p^{h-1}I & p^{h-1}A_{h-1,h} \end{pmatrix}.$$

If the block sizes of the columns are k_0, k_1, \dots, k_h (necessarily summing to n), then

$$|C| = p^k,$$

where

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

Proof

Applying elementary row operations to the matrix does not change the code C generated by the matrix. Since we are also allowed to permute the columns the only impediment to obtaining a generator matrix of the form

$$(I \ B_0),$$

10.4 · Codes over $\mathbb{Z}/p^h\mathbb{Z}$

is rows in which all elements are divisible by p . Thus, we obtain a generator matrix of the form

$$G = \begin{pmatrix} I & B_{01} \\ 0 & pB_{02} \end{pmatrix},$$

for some matrices B_{01} and B_{02} . We continue applying row operations and column permutations. Again, the only impediment to obtaining a generator matrix of the form

$$\begin{pmatrix} I & B_{01} & B_{02} \\ 0 & pI & pB_{12} \end{pmatrix},$$

is rows in which all elements are divisible by p^2 .

Therefore, there is a generator matrix for C of the form

$$G = \begin{pmatrix} I & B_{01} & B_{02} \\ 0 & pI & pB_{12} \\ 0 & 0 & p^2B_{22} \end{pmatrix}.$$

The form of G follows by continuing applying row operations and column permutations. The code generated by G is

$$C = \{(u_1, \dots, u_r)G \mid u_i \in \mathbb{Z}/p^h\mathbb{Z}\}.$$

If all the entries in the ℓ -th row of G are divisible by p^j , then we can restrict u_ℓ to $\mathbb{Z}/p^{h-j}\mathbb{Z}$, which implies that the size of the code is as claimed. \square

Example 10.6

Consider the code over $\mathbb{Z}/8\mathbb{Z}$ generated by the matrix

$$\begin{pmatrix} 0 & 2 & 1 & 4 & 1 & 1 \\ 4 & 6 & 7 & 4 & 7 & 1 \end{pmatrix}.$$

By shifting the coordinates one coordinate to the right, we obtain an equivalent code with generator matrix

$$\begin{pmatrix} 1 & 0 & 2 & 1 & 4 & 1 \\ 1 & 4 & 6 & 7 & 4 & 7 \end{pmatrix}.$$

Subtracting the first row from the second, we obtain a generator matrix for the same code

$$\begin{pmatrix} 1 & 0 & 2 & 1 & 4 & 1 \\ 0 & 4 & 4 & 6 & 0 & 6 \end{pmatrix}.$$

Multiplying the second row by 3 we obtain another generator matrix for the code

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

Finally, interchanging the second and sixth column we obtain an equivalent code with generator matrix

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

Comparing this to the claim of [Theorem 10.5](#), the matrix $A_{01} = (1)$, the matrix $A_{02} = (2 \ 1 \ 4 \ 0)$ and the matrix $A_{12} = (2 \ 1 \ 0 \ 2)$.

Note that the code has size 32 and not 64, which is not immediately apparent from the initial generator matrix. ■

10.5 Codes over $\mathbb{Z}/4\mathbb{Z}$

The **Gray map** is a map γ from $\mathbb{Z}/4\mathbb{Z}$ to $\{0, 1\}^2$ defined by

$$\begin{array}{c|cccc} x & 0 & 1 & 2 & 3 \\ \hline \gamma(x) & (0, 0) & (0, 1) & (1, 1) & (1, 0) \end{array}.$$

We extend the Gray map to a map from $(\mathbb{Z}/4\mathbb{Z})^n$ to $\{0, 1\}^{2n}$ by applying γ to each coordinate.

If C is a block code of length n over $\mathbb{Z}/4\mathbb{Z}$, then $\gamma(C)$, defined by

$$\gamma(C) = \{\gamma(v) \mid v \in C\},$$

is a binary code of length $2n$. It is immediate that if C has minimum distance d , then $\gamma(C)$ has minimum distance at least d .

However, there is a possibility that the minimum distance of $\gamma(C)$ is larger than d .

Example 10.7

Let C be the code over $\mathbb{Z}/4\mathbb{Z}$ generated by the matrix

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

The 8 codewords of C and the code $\gamma(C)$ are

C	$\gamma(C)$
(0,0,0,0,0,0)	(0,0,0,0,0,0,0,0,0,0)
(1,0,2,1,1,1)	(0,1,0,0,1,1,0,1,0,1,0,1)
(0,2,2,2,0,0)	(0,0,1,1,1,1,1,1,0,0,0,0)
(1,2,0,3,1,1)	(0,1,1,1,0,0,1,0,0,1,0,1)
(2,0,0,2,2,2)	(1,1,0,0,0,0,1,1,1,1,1,1)
(2,2,2,0,2,2)	(1,1,1,1,1,1,0,0,1,1,1,1)
(3,0,2,3,3,3)	(1,0,0,0,1,1,1,0,1,0,1,0)
(3,2,0,1,3,3)	(1,0,1,1,0,0,0,1,1,0,1,0)

One readily checks that the minimum distance of C is 3 and the minimum distance of $\gamma(C)$ is 6. ■

The **Lee distance** between two elements u and v of $(\mathbb{Z}/4\mathbb{Z})^n$ is defined as the Hamming distance between $\gamma(u)$ and $\gamma(v)$. The **Lee weight** of an element u of $(\mathbb{Z}/4\mathbb{Z})^n$ is the Lee distance between u and the all zero n -tuple.

Lemma 10.8 *Let C be a linear code over $\mathbb{Z}/4\mathbb{Z}$. The minimum Lee weight of a non-zero codeword of C is equal to the minimum distance of $\gamma(C)$.*

Proof

Let $u = (u_1, \dots, u_n)$ and $v = (v_1, \dots, v_n)$ be two codewords of C .

By checking all possibilities for $u_i, v_i \in \mathbb{Z}/4\mathbb{Z}$, one can verify that the distance between $\gamma(u_i)$ and $\gamma(v_i)$ is equal to the distance between $(0, 0)$ and $\gamma(u_i - v_i)$.

Thus,

$$d(\gamma(u), \gamma(v)) = \sum_{i=1}^n d(\gamma(u_i), \gamma(v_i)) = \sum_{i=1}^n d(\gamma(u_i - v_i), (0, 0))$$

which is equal to the Lee weight of $u - v$. □

In the following example, we return to [Example 10.3](#) and consider the entries in the matrix modulo 4. This matrix will then generate a code over $\mathbb{Z}/4\mathbb{Z}$.

Example 10.9

By [Example 10.3](#), we have that $X^3 + 2X^2 + X + 3$ divides $X^7 - 1$ in $(\mathbb{Z}/4\mathbb{Z})[X]$. This polynomial generates a cyclic code of length 7 which extends to a code of length 8 with generator matrix

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

Let C be the $\mathbb{Z}/4\mathbb{Z}$ -linear code of length 8 with 256 codewords defined by

$$C = \{uG \mid u \in (\mathbb{Z}/4\mathbb{Z})^4\}.$$

The code $\gamma(C)$ is a binary code of length 16 with 256 codewords. By [Exercise 10.7](#), the minimum distance of $\gamma(C)$ is 6. This code is equivalent to the code constructed in [Example 9.12](#). As mentioned there, an important observation is that there is no binary linear code with these parameters, which we proved in [Example 4.20](#). ■

[Example 10.9](#) suggests that codes over rings may be a good place to look for non-linear codes which have better parameter sets than linear codes. It may be the case that we need to consider non-linear codes to disprove [Conjecture 3.18](#).

10.6 Comments

This chapter leans somewhat on the enlightening article by Calderbank and Sloane on p -adic codes [14]. Carlet [19] has generalised the Gray map to a bijection from $\mathbb{Z}/2^k\mathbb{Z}$ to $R(1, k-1)$. This can be extended to $(\mathbb{Z}/2^k\mathbb{Z})^n$ and can therefore be used to construct (non-linear) binary codes from $\mathbb{Z}/2^k\mathbb{Z}$ -linear codes.

[Theorem 10.2](#) is a special case of Hensel's lifting lemma. For more on p -adic numbers, including the lifting lemma, see [30].

10.7 Exercises

10.1 Let

$$\lambda = (1, b_2, b_3, b_4, \dots)$$

be the 2-adic integer which is a root of $X^2 - X + 2$. Calculate the numbers b_2, b_3, b_4 in the sequence of λ .

10.2 Prove that the code generated by the 4×8 matrix obtained by extending the generator matrix in [Example 10.3](#) with the all-one vector is a self-dual code.

10.3 Check, with the aid of a computer or not, that the code in [Example 10.4](#) has minimum distance 11.

10.4 i. Prove that the dual code C^\perp , to the code C generated by the matrix in [Theorem 10.5](#), has a generator matrix of the form

$$G = \begin{pmatrix} B_{0,h} & B_{0,h-1} & \cdots & B_{03} & B_{02} & B_{01} & I \\ pB_{1,h} & pB_{1,h-1} & \cdots & pB_{13} & pB_{12} & pI & 0 \\ p^2B_{2,h} & p^2B_{2,h-1} & \cdots & p^2B_{23} & p^2I & 0 & 0 \\ \cdot & \cdot & \cdot & \ddots & \ddots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \ddots & \ddots & \cdot & \cdot \\ p^{h-1}B_{h-1,h} & p^{h-1}I & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix},$$

for some matrices B_{ij} , where the blocks of columns have the same size as in [Theorem 10.5](#).

ii. Prove that $|C^\perp| = p^{k_\perp}$, where

$$k_\perp = \sum_{i=1}^h ik_i.$$

10.5 Let C be a linear code over $\mathbb{Z}/p^h\mathbb{Z}$. Prove that $(C^\perp)^\perp = C$.

10.6 Let C be the linear code over $\mathbb{Z}/4\mathbb{Z}$ from [Example 10.7](#).

- Check that the minimum Lee weight of a non-zero codeword of C is 6 and verify that the minimum Hamming distance between any two codewords of $\gamma(C)$ is 6.
- The code $\gamma(C)$ is a non-linear binary code of length 12, minimum distance 6 and size 8. Construct a linear code with the same parameters.
- The code

$$C = \{\lambda u + 2\mu v \mid \lambda \in \mathbb{Z}/4\mathbb{Z}, \mu \in \mathbb{Z}/2\mathbb{Z}\}$$

for some $u \in (\mathbb{Z}/4\mathbb{Z})^6$ and $v \in (\mathbb{Z}/2\mathbb{Z})^6$, where the weight of v is 3. Construct a code with the same parameters as C in which the weight of v is 4.

10.7

- Prove, using row operations, that the code C in [Example 10.9](#) has a generator matrix

$$G = G_1 + 2G_2,$$

where

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

and

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

ii. Prove, using [Lemma 10.8](#), that the code $\gamma(C)$ in [Example 10.9](#) has minimum distance 6.

10.8

- i. Prove that $X^2 + \lambda X - 1$ divides $X^8 - 1$ in $\mathbb{Z}_p[X]$, where λ is a p -adic integer satisfying $\lambda^2 = -2$.
- ii. Calculate the next few numbers in the sequences $(1, 4, \dots)$ and $(2, 5, \dots)$ which are both solutions of $\lambda^2 = -2$ in \mathbb{Z}_3 .

10.9

- i. Prove that $X^5 + \lambda X^4 - X^3 + X^2 + (\lambda - 1)X - 1$ divides $X^{11} - 1$ in $\mathbb{Z}_p[X]$, where λ is a p -adic integer satisfying $\lambda^2 = \lambda - 3$.
- ii. Calculate the first few numbers in the sequences which are solutions of $\lambda^2 = \lambda - 3$ in \mathbb{Z}_3 .

10.10

- i. Prove that $X^{11} + \lambda X^{10} + (\lambda - 3)X^9 - 4X^8 - (\lambda + 3)X^7 - (2\lambda + 1)X^6 - (2\lambda - 3)X^5 - (\lambda - 4)X^4 + 4X^3 + (\lambda + 2)X^2 + (\lambda - 1)X - 1$ divides $X^{23} - 1$ in $\mathbb{Z}_p[X]$, where λ is a p -adic integer satisfying $\lambda^2 = \lambda - 6$.
- ii. Calculate the first few numbers in the sequences which are solutions of $\lambda^2 = \lambda - 6$ in \mathbb{Z}_2 .

Hints and Answers to Selected Exercises

1.1 $H_r(X) = \log_r(2^{23/18}3^{5/3}5^{-5/18}) \approx \log_r 9.6759.$

1.2 $H_2(X) = n - \log_2(2^n - 1) + (1 - 2^{-n})(2 - 2^{1-n} - n2^{-n}).$

1.3

- i. $\binom{n}{j}2^{n-j}.$
- ii. $H_2(X) = \frac{3}{2}.$

1.4 $\phi h_r(p).$

1.5 $H_r(Y) = m(1 - \log_r m),$ where $m = (\phi r + 1 - 2\phi)/(r - 1).$
 $I(X, Y) = m(1 - \log_r m) + \phi \log_r \phi + r \log_r((1 - \phi)/(r - 1)).$

1.6 $\phi.$

1.8

- i. $H(Y|X) = -(1 - p)(\frac{3}{4} \log_r 3 - \log_r 4).$
 $H(Y) = h_r(p) - (1 - p)(\frac{3}{4} \log_r 3 - \log_r 4).$
 $H(X) = h_r(p).$
- ii. $\log_r 2.$

1.9

- i. $-\frac{3}{4}p \log 3 + \frac{3}{2}p \log 2 - \left(\frac{4-3p}{4}\right) \log \left(\frac{4-3p}{8}\right).$
- iii. $I(X, Y)$ evaluated at $p = 4/(2^{8/3} + 3).$

2.5

- i. 1, 2, 4, 4, 4.
- ii. 1, 11, 11.
- iii. 1, 1, 1, 1, 2, 2, 2.

2.6

- i. $(X - 1)(X - 2)(X - 4)(X^3 - 2)(X^3 - 4).$
- ii. $(X - 1)(X^{10} + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1).$
- iii. $(X - 1)(X + 1)(X^2 + 1)(X^2 + 5X - 1)(X^2 - 5X - 1).$

2.12 Hint: Let U be a k -dimensional subspace of \mathbb{F}_q^{k+1} , so U is a hyperplane of $\text{PG}(k, q)$. Let V be an $(r+1)$ -dimensional subspace of \mathbb{F}_q^{k+1} and observe that $v+(V \cap U)$, where $v \in V \setminus U$, is a coset of a subspace of $U \cong \mathbb{F}_q^k$.

3.4 Hint: Assuming $(0, 0, 0, 0, 0, 0, 0, 0, 0) \in C$, the non-zero vectors all have weight 6. Look at the proof of [Lemma 3.10](#).

The codewords are the six rows of the matrix

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

3.5 Hint: By considering the proof of [Lemma 3.10](#), observe that equality in the bound cannot occur.

3.6 Hint: follow the same strategy as [Exercise 3.5](#).

3.12

- ii. The vector $\sigma(u)$ has $n - \text{wt}(u)$ ones and $\text{wt}(u)$ minus ones, where $\text{wt}(u)$ is the number of ones that u has.
- iii. If u and v agree on a coordinate, then that coordinate contributes 1 to the scalar product $\sigma(u) \cdot \sigma(v)$. If u and v differ on a coordinate, then that coordinate contributes -1 to the scalar product $\sigma(u) \cdot \sigma(v)$. They agree on $n - d(u, v)$ coordinates and differ on $d(u, v)$ coordinates.
- iv. Using ii. and iii. and the fact that $d(u, v) \geq d$ for all $u, v \in C$, the bound follows.
- v. The bound on w ensures that the scalar product in iv. is non-positive, with $\lambda = \frac{1}{2}\sqrt{1 - 2(d/n)}$. Assuming that $(0, 0, \dots, 0) \in C$, we have by i. and iv. that the number of codewords at a distance at most w from $(0, 0, \dots, 0)$ is at most $2n$.

4.3 Prove there exists a $[n - k + r, r, d]_q$ code for $r = 1, \dots, k$, by induction on r . Note that there is a $[n - k + 1, 1, d]_q$ code, since the condition implies $d \leq n - k + 1$. The condition implies the condition for n replaced by $n - 1$ and k replaced by $k - 1$ so, by induction, there exists a $[n - 1, k - 1, d]_q$ code. Let H be a $(n - k) \times (n - 1)$ check matrix for this code. The condition allows us to add a column to H so that any $d - 1$ columns of the extended matrix are linearly independent, i.e. there is a non-zero vector of \mathbb{F}_q^{n-k} which is not a linear combination of any subset of at most $d - 2$ columns of H . Apply [Lemma 4.4](#).

4.4 Hint: Since the code is self-dual it has no codewords of odd weight.

The set of equations in matrix form is

$$\begin{pmatrix} -15 & 1 & 1 & 1 & 1 \\ 8 & 4 & 0 & -4 & -8 \\ 28 & -12 & -4 & 4 & 8 \\ 56 & -4 & 0 & 4 & -56 \\ 70 & -10 & -10 & -10 & 70 \\ 56 & -4 & 0 & 4 & -56 \\ 28 & 4 & -4 & -12 & 8 \\ 8 & 4 & 0 & -4 & -8 \\ 1 & 1 & 1 & 1 & -15 \end{pmatrix} \begin{pmatrix} 1 \\ a_2 \\ a_4 \\ a_6 \\ a_8 \end{pmatrix} = 0.$$

4.5 The sent vector is $(1, 1, 1, 0, -1, -1, -1, 0)$.

4.6 $A(X) = 1 + 4X^3 + 3X^4$.

4.7 The sent vector is $(1, 2, 3)G$.

4.8

- i. Prove that every 4×4 sub-matrix of H is non-singular.
- ii. The sent vector is $(1, 6, 3, 6, 1, 2, 2)$.

4.9 The sent vector is $(1, 2, 1, 2, 0, 2, 0, 2, 0)$.

4.14 For example,

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

4.15 Let u and v be two codewords of weight w with the same support. For all non-zero $\lambda \in \mathbb{F}_q$ the vector λv is a codeword. For each non-zero coordinate u_i of u there is a λ such that $u_i = \lambda v_i$, where v_i is the i -th coordinate of v . By the pigeon-hole principle, there is a λ such that $\lceil w/(q-1) \rceil$ of the coordinates agree with the corresponding coordinate in λv . Then $u - \lambda v$ is a codeword of weight at most $w - \lceil w/(q-1) \rceil$.

4.16

- i. Hint: Make the substitution $X \rightarrow 1 - X$ in the MacWilliams identities.
- ii. Hint: The subspace of polynomials spanned by

$$\{X^j(1 + (q-1)(1-X))^{n-j} \mid j = d, \dots, n\}$$

is equal to the subspace of polynomials spanned by

$$\{X^j \mid j = d, \dots, n\}.$$

- iii. Applying ii., there is a unique way to write the polynomial $(1 + (q - 1)(1 - X))^n$ as a linear combination of the $n + 1$ polynomials in ii.
 iv. Apply [Theorem 4.13](#).

5.1 Hint: Since C is self-dual, all codewords have weight which are multiples of 3. Using this observation, solve the equations given by the equality

$$729A(X) = (1 + 2X)^{12}A\left(\frac{1 - X}{1 + 2X}\right),$$

given by [Theorem 4.13](#).

5.2 $\overleftarrow{g}(X) = X^{11} + X^{10} + X^6 + X^5 + X^4 + X^2 + 1.$

5.6 The largest dimensions are i. 7. ii. 11 iii. 4.

5.7

- i. $f(X) = X^8 + X^7 + X^6 + X^4 + X^2 + X + 1.$
 $g(X) = X^8 + X^5 + X^4 + X^3 + 1.$
 ii. Use [Theorem 5.10](#).
 iii. Use [Theorem 3.5](#) and prove that the extended code is also linear.

5.8

- i. The polynomial $X^{11} + 1$ factorises as

$$(X + 1)(X^5 + \epsilon X^4 + X^3 + X^2 + \epsilon^2 X + 1)(X^5 + \epsilon^2 X^4 + X^3 + X^2 + \epsilon X + 1),$$

where $\epsilon^2 = 1 + \epsilon.$

- iii. Hint: Use [Exercise 5.5](#).

5.9

- ii. Apply [Exercise 5.5](#).
 iv. Find a codeword of weight 7.

6.1 Hint: use [Exercise 2.8](#).

6.2 Hint: $(c_0, \dots, c_{n-1}) \in C$ if and only if there is a polynomial h of degree at most $k - 1$ such that $c_i = h(\alpha^i)$. Consider the polynomial $c(X) = \sum_{i=0}^{n-1} c_i X^i$. To prove the exercise observe that it is sufficient to prove that $c(\alpha^j) = 0$, for all $j = 1, \dots, n - k$.

6.3 The sent codeword is the evaluation of the polynomial $F(X) = X^3 - X + 1.$

6.5 Hint: The 4×4 sub-matrices are Vandermonde.

6.6 Hint: use the Griesmer bound.

6.7 Hint: use i. to prove ii.

6.8

- ii. Use [Exercise 6.7](#) iii.
 iii. For q odd, use [Exercise 6.7](#) ii. For q even, use ii.

6.9 Hint: $x^{2^e-1} = y^{2^e-1}$ implies $x = y$.

6.10 Hint: Let A be a 5×5 sub-matrix of the matrix. Prove that $(\det A)^3 = v - \eta v^3$, where v is the determinant of a Vandermonde matrix. Apply [Theorem 6.10](#).

6.11 Hint: Prove that $\det A \neq 0$ and use [Theorem 6.10](#).

7.1 The sent codeword is $(-1, 1, 1, 1, 1, 0, 1, 1, 1)$.

7.3 Hint: Use [Exercise 7.2](#).

7.7 $g(X) = aX^2$, $h(X) = aX$, $f(X) = X$.

7.9 $g(X) = (X + 1)h(X)$, $h(X) = a(2(e + 1)X + 1)$, $f(X) = X + 1$.

7.10

- i. $(Y/Z) = 4P_3 - 4P_2$.
 ii. $\{1, X/Y, X/Z, (Y^2 + Z^2)/X^2\}$.

8.2

- iii. 1.

8.5

- iii. The minimum distance is 6.
 iv. The dimension is 11.

9.1 The sent vector is

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

which the evaluation of the polynomial $X_3 + X_4 + X_1X_2$.

9.2 Hint: Start by selecting a vector e_1 such that $q(e_1) = 0$. Then choose e_2 such that $q(e_2) = 0$ and $b(e_1, e_2) = 0$, where $b(X, Y) = q(X + Y) - q(X) - q(Y)$.

9.5 Hint: we know the first row in the remaining matrices. Since the matrices are symmetric and have zeroes on the diagonal, each of the remaining matrices has only three entries to be determined.

The four additional matrices are

$$\left\{ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \right\}$$

9.6

- i. As in the proof of [Lemma 9.9](#).
 ii. As in the proof of [Lemma 9.10](#) but there is also the case that the $q(X) + \ell(X)$ is of the form

$$\sum_{i=1}^r X_{2i-1} X_{2i} + X_{2r+1}$$

after a suitable change of basis.

- iv. Hint: Find a basis for a 5-dimensional subspace of five 5×5 matrices all of whose non-zero matrices have rank at least 4.

9.7 The number of monomials of degree i in m indeterminates in which the degree of each indeterminate is at most $q - 1$ is equal to the number of solutions of $x_1 + \cdots + x_m = i$, where $0 \leq x_j \leq q - 1$. Use inclusion-exclusion.

10.1 Hint: $b_2 = 2c_2 + 1$ for some $c_2 \in \{0, 1\}$. We can solve for c_2 from

$$(2c_2 + 1)^2 - (2c_2 + 1) + 2 = 0 \pmod{4}.$$

$$\lambda = (1, 3, 3, 11, \dots).$$

10.6

- ii. The linear code over \mathbb{F}_2 generated by the matrix

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

has minimum distance 6.

- iii. For example, the code generated by the matrix

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

10.8

- ii. $(1, 4, 22, 76, \dots)$ and $(2, 5, 5, 59, \dots)$.

10.9

- ii. $(0, 3, 12, \dots)$ and $(1, 7, 16, \dots)$.

10.10

- ii. $(0, 2, 2, 10, \dots)$ and $(1, 3, 7, 7, \dots)$.

Bibliography

1. T. Alderson, A. Gács, On the maximality of linear codes. *Des. Codes Cryptogr.* **53**, 59–68 (2009)
2. T. Alderson, A.A. Bruen, R. Silverman, Maximum distance separable codes and arcs in projective spaces. *J. Combin. Theory Ser. A* **114**, 1101–1117 (2007)
3. R.B. Ash, *Information Theory* (Dover, Mineola, 1965)
4. E.F. Assmus Jr., H.F. Mattson Jr., New 5-designs. *J. Combin. Theory* **6**, 122–151 (1969)
5. S. Ball, On sets of vectors of a finite vector space in which every subset of basis size is a basis. *J. Eur. Math. Soc.* **14**, 733–748 (2012)
6. S. Ball, *Finite Geometry and Combinatorial Applications*. London Mathematical Society Student Texts, vol. 82 (Cambridge University Press, Cambridge, 2015)
7. L.A. Bassalygo, New upper bounds for error-correcting codes. *Probl. Inform. Transm.* **1**, 32–35 (1965)
8. E.R. Berlekamp, *Algebraic Coding Theory* (McGraw-Hill, New York, 1968)
9. E.R. Berlekamp, L.R. Welch, Error correction for algebraic block codes, U.S. Patent 4,633,470, 30 Dec 1986
10. J. Bierbrauer, *Introduction to Coding Theory*, 2nd edn. (Chapman and Hall/CRC Press, Boca Raton, 2016)
11. A. Bishnoi, Some contributions to incidence geometry and the polynomial method. PhD thesis, Universiteit Gent, Gent, 2016
12. R.E. Blahut, The Gleason-Prange theorem. *IEEE Trans. Inform. Theory* **37**, 1269–1273 (2006)
13. R.C. Bose, D.K. Ray-Chaudhuri, On a class of error correcting binary group codes. *Inform. Control* **3**, 68–79 (1960)
14. R. Calderbank, N.J.A. Sloane, Modular and p -adic cyclic codes. *Des. Codes Cryptogr.* **6**, 21–35 (1995)
15. R. Calderbank, E.M. Rains, P.W. Shor, N.J.A. Sloane, Quantum error correction via codes over $GF(4)$. *IEEE Trans. Inform. Theory* **44**, 1369–1387 (1998)
16. P.J. Cameron, *Combinatorics: Topics, Techniques, Algorithms* (Cambridge University Press, Cambridge, 1994; reprinted 1996)
17. P.J. Cameron, J.H. van Lint, *Designs, Codes, Graphs and Their Links*. London Mathematical Society Student Texts, vol. 22 (Cambridge University Press, Cambridge, 1991)
18. T. Can, N. Rengaswamy, R. Calderbank, H.D. Pfister, Kerdock codes determine unitary 2-designs (2019), <https://arxiv.org/abs/1904.07842>
19. C. Carlet, \mathbb{Z}_{2^k} -linear codes. *IEEE Trans. Inform. Theory* **44**, 1543–1547 (1998)
20. J. Davis, IMS Public Lecture: Apple vs Samsung: a Mathematical Battle, <https://ims.nus.edu.sg/resourcevideo.php>, 18 May 2016
21. P. Delsarte, *An Algebraic Approach to the Association Schemes of Coding Theory*. Philips Research Reports Supplement, vol. 10 (N.V. Philips' Gloeilampenfabrieken, Amsterdam, 1973)
22. L.E. Dickson, *Linear Groups: With an Exposition of the Galois Field Theory* (Dover Publications, New York, 1901)
23. R. Fano, *Transmission of Information; A Statistical Theory of Communications* (MIT Press, Cambridge, 1961)
24. R.G. Gallager, *Low Density Parity Check Codes* (MIT Press, Cambridge, 1963)
25. E.N. Gilbert, A comparison of signalling alphabets. *Bell Syst. Tech. J.* **31**, 504–522 (1952)
26. D.G. Glynn, The non-classical 10-arc of $PG(4, 9)$. *Discrete Math.* **59**, 43–51 (1986)
27. M.J.E. Golay, Notes on digital coding. *Proc. Inst. Radio Eng.* **37**, 657 (1949)

28. V.D. Goppa, A new class of linear error-correcting codes. *Probl. Inform. Transm.* **6**, 207–212 (1970)
29. V.D. Goppa, Codes on algebraic curves. *Soviet Math. Dokl.* **24**, 170–172 (1981)
30. F. Gouvea, *p-Adic Numbers: An Introduction* (Springer, Berlin, 1997)
31. J.H. Griesmer, A bound for error-correcting codes. *IBM J. Res. Dev.* **4**, 532–542 (1960)
32. V. Guruswami, A. Rudra, Limits to list decoding Reed-Solomon codes. *IEEE Trans. Inform. Theory* **52**, 3642–3649 (2006)
33. V. Guruswami, M. Sudan, Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory* **45**, 1757–1767 (1999)
34. R.W. Hamming, Error detecting and error correcting codes. *Bell Labs Tech. J.* **29**, 147–160 (1950)
35. R.V.L. Hartley, Transmission of information. *Bell Syst. Tech. J.* **7**, 535–563 (1928)
36. H.J. Helgert, Alternant codes. *Inform. Control* **26**, 369–380 (1974)
37. R. Hill, *A First Course in Coding Theory* (Oxford University Press, Oxford, 1988)
38. A. Hocquenghem, Codes correcteurs d’erreurs. *Chiffres* **2**, 147–156 (1959)
39. S. Hoory, N. Linial, A. Wigderson, Expander graphs and their applications. *Bull. Am. Math. Soc.* **43**, 439–561 (2006)
40. G.A. Jones, J.M. Jones, *Information and Coding Theory*. Springer Undergraduate Mathematics Series (Springer, Berlin, 2000)
41. W.M. Kantor, An exponential number of generalized Kerdock codes. *Inform. Control* **53**, 74–80 (1982)
42. W.M. Kantor, Spreads, translation planes and Kerdock sets, I, II. *SIAM J. Algebraic Discrete Math.* **3**, 151–165 and 308–318 (1983)
43. T. Kasami, S. Lin, W.W. Peterson, Generalized Reed-Muller codes. *Electron. Commun. Jpn.* **51**, 96–104 (1968)
44. A.M. Kerdock, A class of low-rate non-linear binary codes. *Ann. Univ. Turku, Ser. A* **20**, 182–187 (1972)
45. Y. Kou, S. Lin, M. Fossorier, Low-density parity-check codes based on finite geometries: a rediscovery and new results. *IEEE Trans. Inform. Theory* **47**, 2711–2736 (2001)
46. R. Lidl, H. Niederreiter, *Finite Fields*. Encyclopedia of Mathematics and Its Applications, vol. 20, 2nd edn. (Cambridge University Press, Cambridge, 1997)
47. S. Lin, E.J. Weldon, Long BCH codes are bad. *Inform. Control* **11**, 445–451 (1967)
48. S. Ling, C. Xing, *Coding Theory: A First Course* (Cambridge University Press, Cambridge, 2004)
49. D.J.C. MacKay, R.M. Neal, Near Shannon limit performance of low density parity check codes. *Electron. Lett.* **32**, 1645–1646 (1996)
50. F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes* (North-Holland, New York, 1977)
51. R.J. McEliece, E.R. Rodemich, H. Rumsey, L.R. Welch, New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Trans. Inform. Theory* **23**, 157–166 (1997)
52. G.L. Mullen, D. Panario (eds.), *Handbook of Finite Fields*. Discrete Mathematics and Its Applications (CRC Press, Boca Raton, 2013)
53. D.E. Muller, Application of Boolean algebra to switching circuit design and to error detection. *IEEE Trans. Comput.* **3**, 6–12 (1954)
54. A.W. Nordstrom, J.P. Robinson, An optimum nonlinear code. *Inform. Control* **11**, 613–616 (1967)
55. P.R.J. Östergård, On binary/ternary error-correcting codes with minimum distance 4, in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, ed. by M. Fossorier, H. Imai, S. Lin, A. Poli. Lecture Notes in Computer Science, vol. 1719 (Springer, Berlin, 1999), pp. 472–481
56. T. Penttilä, I. Pinneri, Hyperovals. *Australas. J. Combin.* **19**, 101–114 (1999)
57. V. Pepe, LDPC codes from the Hermitian curve. *Des. Codes Cryptogr.* **42**, 303–315 (2007)
58. M. Plotkin, Binary codes with specified minimum distance. *IRE Trans. Inform. Theory* **6**, 445–450 (1960)
59. I.S. Reed, A class of multiple-error-correcting codes and the decoding scheme. *IEEE Trans. Inform. Theory* **4**, 38–49 (1954)

60. I.S. Reed, G. Solomon, Polynomial codes over certain finite fields. *J. Soc. Ind. Appl. Math.* **8**, 300–304 (1960)
61. S. Roman, *Coding and Information Theory*. Graduate Texts in Mathematics, vol. 134 (Springer, Berlin, 1992)
62. C. Roos, A note on the existence of perfect constant weight codes. *Discrete Math.* **47**, 121–123 (1983)
63. B. Segre, Ovals in a finite projective plane. *Canad. J. Math.* **7**, 414–416 (1955)
64. B. Segre, Introduction to Galois geometries. *Atti Accad. Naz. Lincei Mem.* **8**, 133–236 (1967)
65. C.E. Shannon, *A Mathematical Theory of Communication* (University of Illinois Press, Champaign, 1949; reprinted 1998)
66. R.C. Singleton, Maximum distance q -nary codes. *IEEE Trans. Inform. Theory* **10**, 116–118 (1964)
67. M. Sipser, D.A. Spielman, Expander codes. *IEEE Trans. Inform. Theory* **42**, 1710–1722 (1996)
68. M. Sudan, Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity* **13**, 180–193 (1997)
69. A. Ta-Shma, Explicit, almost optimal, epsilon-balanced codes, in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (2017), pp. 238–251
70. C. Thommesen, The existence of binary linear concatenated codes with Reed-Solomon outer codes which asymptotically meet the Gilbert-Varshamov bound. *IEEE Trans. Inform. Theory* **29**, 850–853 (1983)
71. M.A. Tsfasman, Algebraic-Geometric codes and asymptotic problems. *Discrete Applied Math.* **33**, 241–256 (1991)
72. M.A. Tsfasman, S.V. Vlăduț, *Algebraic-Geometric Codes* (Kluwer Academic Publishers, Norwell, 1991)
73. M.A. Tsfasman, S.V. Vlăduț, T. Zink, Modular curves, Shimura curves, and Goppa codes, better than the Varshamov-Gilbert bound. *Math. Nachr.* **109**, 21–28 (1982)
74. J.H. van Lint, *Introduction to Coding Theory*. Graduate Texts in Mathematics, vol. 86, 3rd edn. (Springer, Berlin, 1999)
75. R.R. Varshamov, Estimate of the number of signals in error correcting codes. *Dokl. Acad. Nauk SSSR* **117**, 739–741 (1957)
76. S.B. Wicker, V.K. Bhargava (eds.), *Reed-Solomon Codes and Their Applications* (IEEE Press, Piscataway, 1994)

Index

A

$A(n, d, w)$, 38, 44
 $A_r(n, d)$, 32, 43
 Affine plane, 26, 131
 Affine space, 26
 $AG(k, q)$, 26, 131
 Alderson-Bruen-Silverman model, 59
 Alphabet, 29
 Arc, 92

B

Belief propagation, 128
 Block code, 10
 Boolean function, 133
 Bound

- Elias–Bassalygo, 39
- Gilbert–Varshamov, 32, 67, 109
- Griesmer, 60, 69
- linear programming, 66
- McEliece–Rodemich–Rumsey–Welch, 41
- Plotkin, 35, 37, 63
- Singleton, 84
- sphere packing, 33, 36, 44, 63

 Burst errors, 94

C

Channel, 4

- binary erasure, 6, 15
- binary symmetric, 5, 11
- capacity of, 11
- information, 4

 Character, 55
 Check matrix, 48, 51, 106, 120, 126
 Code

- algebraic geometric, 112
- alternant, 107
- asymptotically good, 36
- BCH, 78, 81
- block, 10, 29
- constant weight, 38
- cyclic, 71, 156

- dual, 54, 72, 136, 155, 162
- equivalent, 59
- evaluation, 85
- extended, 31, 67
- extension, 31
- generalised Reed–Solomon, 107, 120
- Golay, 73, 76, 80, 81
- Hamming, 49
- Kerdock, 142, 144
- LDPC, 126
- length of, 29
- linear, 47, 92, 155, 157
- local reconstruction, 101
- MDS, 83
- Nordstrom–Robinson, 145
- over a ring, 157
- over $\mathbb{Z}/4\mathbb{Z}$, 160, 163
- p -adic, 155
- perfect, 34, 43, 67, 73
- quadratic residue, 75
- Reed–Muller, 133
- Reed–Solomon, 85, 96, 101
- repetition, 30
- self-dual, 54
- subfield sub, 105
- systematic, 38
- turbo, 131

 Combinatorial design, 63
 Conjecture

- constant weight binary codes, 42
- Gilbert–Varshamov, 42
- MDS, 95

 Coordinate ring, 113
 Cyclotomic polynomial, 21, 71, 81

D

Decoding, 10

- belief propagation, 128
- list, 88
- majority-logic, 138
- maximum likelihood, 10
- nearest neighbour, 30

- Reed–Solomon code, 86
- standard array, 88
- syndrome, 51, 67

Degree

- of a divisor, 114

Design, 63

Distance

- Hamming, 11, 30
- Lee, 161
- minimum, 30

Divisor, 113

- degree of, 114

Dual code, 54, 72

E

End-vertex, 123

Entropy, 3

- conditional, 6
- input, 5
- joint, 6
- output, 5

Entropy function, 3

Expander graph, 124

F

Field, 17

- finite, 19
- subfield, 26

Frobenius automorphism, 21, 55

G

Generator matrix, 48, 69, 155, 157

Genus, 114

Graph, 123

- bipartite, 123

Gray map, 160

H

Hamming distance, 11

 $h(p)$, 3, 12, 32

Hyperplane, 24

I

Information, 2

- average, 6
- mutual, 8

L

Lee distance, 161

Lee weight, 161

List decoding, 88

M

MacWilliams identities, 56

Matrix

- check, 48, 51, 106, 120, 126
- generator, 48, 69, 155, 157

Minimum distance, 47, 161

- prescribed, 78, 115

Minimum weight, 47

Module, 155

N

Neighbour, 123

 $[n, k, d]_q$, 48 $(n, K, d)_r$, 48**P** p -adic integers, 151 p -adic numbers, 152 $PG(k-1, q)$, 24, 92

Plotkin lemma, 34, 43

Polynomial

- cyclotomic, 21
- interpolation, 99, 108
- Krawtchouk, 66

Primitive element, 22

Probability

- backward, 4
- of a correct decoding, 10
- forward, 4
- joint, 4

Projective plane, 24

Projective space, 24, 92

R

Random variable, 1

Rate, 10, 36

Ring, 17

- coordinate, 113

S

Shannon's theorem, 13
Stable set, 123
Standard array decoding, 88
Submodule, 155
Support, 63
Syndrome, 51, 128
Syndrome decoding, 51, 67

T

Trace map, 55
Transmission rate, 10

W

Weight, 37, 47, 71
– Lee, 161
Weight enumerator, 54