# CONCISE ENCYCLOPEDIA OF CODING THEORY

Edited by
W. Cary Huffman
Jon-Lark Kim
Patrick Solé

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

# Concise Encyclopedia of Coding Theory

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

# Concise Encyclopedia of Coding Theory

Edited by

W. Cary Huffman
Loyola University Chicago, USA

Jon-Lark Kim
Sogang University, Republic of Korea

Patrick Solé
University Aix-Marseille, Marseilles, France.

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

First edition published 2021
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN



CRC Press is an imprint of Taylor & Francis Group, LLC



ISBN: 9781138551992 (hbk)
ISBN: 9781315147901 (ebk)

To a new generation, my grandchildren

*Ezekiel James Huffman* – 20 November 2019

*Eliana Rei Beaton* – 24 June 2020

– W. C. H.

To my daughter *Sylvie Jinna Kim*

To the memory of my advisor

*Vera Pless* – 1931–2020

– J.-L. K.

To the memory of my advisor

*Gérard D. Cohen* – 1951–2018

– P. S.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

# *Contents*

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

# *Preface*

The 1948 publication of the paper *A mathematical theory of communication* by Claude E. Shannon is considered to be the genesis of the now vast area of coding theory. In the over 70 years since this monumental work first appeared, coding theory has grown into a discipline intersecting mathematics, computer science, and engineering with applications to almost every area of communication and data storage and even beyond. Given a communication channel on which data is transmitted or a storage device on which data is kept, that data may be corrupted by errors or erasures. In what form do you put that data so that the original information can be recovered and how do you make that recovery? Shannon's paper showed that coding theory provides an answer to that question. The *Concise Encyclopedia of Coding Theory*, somewhat in the spirit of the 1998 *Handbook of Coding Theory* [1521], examines many of the major areas and themes of coding theory taking the reader from the basic introductory level to the frontiers of research.

The authors chosen to contribute to this encyclopedia were selected because of their expertise and understanding of the specific topic of their chapter. Authors have introduced the topic of their chapter in relationship to how it fits into the historical development of coding theory and why their topic is of theoretical and/or applied interest. Each chapter progresses from basic to advanced ideas with few proofs but with many references to which the reader may go for more in-depth study. An attempt has been made within each chapter to point the reader to other chapters in the encyclopedia that deal with similar or related material. An extensive index is provided to help guide the reader interested in pursuing a particular concept.

The *Concise Encyclopedia of Coding Theory* is divided into of three parts: Part I explores the fundamentals of coding theory, Part II examines specific families of codes, and Part III focuses on the practical application of codes.

The first thirteen chapters make up Part I of this encyclopedia. This part explores the fundamental concepts involved in the development of error-correcting codes. Included is an introduction to several historically significant types of codes along with some of their natural generalizations. The mathematical theory behind these codes and techniques for studying them are also introduced. Readers of this encyclopedia who are new to coding theory are encouraged to first read Chapter 1 and then consider other chapters that interest them. More advanced readers may wish to skim Chapter 1 but then move to other chapters.

Chapter **1**, written by the editors of this encyclopedia, is an introduction to the *basic concepts of coding theory* and sets the stage with notation and terminology used throughout the book. The chapter starts with a simple communication channel moving then to the definition of linear and nonlinear codes over finite fields. Basic concepts needed to explore codes are introduced along with families of classical codes: Hamming, Reed–Muller, cyclic, Golay, BCH, and Reed–Solomon codes. The chapter concludes with an brief introduction to encoding, decoding, and Shannon's Theorem, the latter becoming the justification and motivation for the entire discipline.

Chapter **2**, written by Cunsheng Ding, describes two fundamental constructions of *cyclic codes* and the BCH and Hartmann–Tzeng Bounds on cyclic codes. The main task of this chapter is to introduce several important families of cyclic codes, including irreducible cyclic

codes, reversible cyclic codes, BCH codes, duadic codes, punctured generalized Reed–Muller codes, and a new generalization of the punctured binary Reed–Muller codes.

Shannon's proof of the existence of good codes is non-constructive and therefore of little use for applications, where one needs one or all codes with specific (small) parameters. Techniques for *constructing and classifying codes* are considered in Chapter **3**, written by Patric R. J. Östergård, with an emphasis on computational methods. Some classes of codes are discussed in more detail: perfect codes, MDS codes, and general binary codes.

*Self-dual codes* form one of the important classes of linear codes because of their rich algebraic structure and their close connections with other combinatorial configurations like block designs, lattices, graphs, etc. Topics covered in Chapter **4**, by Stefka Bouyuklieva, include construction methods, results on enumeration and classification, and bounds for the minimum distance of self-dual codes over fields of size 2, 3, and 4.

*Combinatorial designs* often arise in codes that are optimal with respect to certain bounds and are used in some decoding algorithms. Chapter **5**, written by Vladimir D. Tonchev, summarizes links between combinatorial designs and perfect codes, optimal codes meeting the restricted Johnson Bound, and linear codes admitting majority logic decoding.

Chapters 1–5 explore codes over fields; Chapter **6**, by Steven T. Dougherty, introduces *codes over rings*. The chapter begins with a discussion of quaternary codes over the integers modulo 4 and their Gray map, which popularized the study of codes over more general rings. It then discusses codes over rings in a very broad sense describing families of rings, the Chinese Remainder Theorem applied to codes, generating matrices, and bounds. It also gives a description of the MacWilliams Identities for codes over rings.

Quasi-cyclic codes form an important class of algebraic codes that includes cyclic codes as a special subclass. Chapter **7**, coauthored by Cem Güneri, San Ling, and Buket Özkaya, focuses on the algebraic structure of *quasi-cyclic codes*. Based on these structural properties, some asymptotic results, minimum distance bounds, and further applications, such as the trace representation and characterization of certain subfamilies of quasi-cyclic codes, are elaborated upon.

Cyclic and quasi-cyclic codes are studied as ideals in ordinary polynomial rings. Chapter **8**, by Heide Gluesing-Luerssen, is a survey devoted to *skew-polynomial rings and skew-cyclic block codes*. After discussing some relevant algebraic properties of skew polynomials, the basic notions of skew-cyclic codes, such as generator polynomial, parity check polynomial, and duality are investigated. The basic tool is a skew-circulant matrix. The chapter concludes with results on constructions of skew-BCH codes.

The coauthors Jürgen Bierbrauer, Stefano Marcugini, and Fernanda Pambianco of Chapter **9** develop the theory of *cyclic additive codes*, both in the permutation sense and in the monomial sense when the code length is coprime to the characteristic of the underlying field. This generalizes the classical theory of cyclic and constacyclic codes, respectively, from the category of linear codes to the category of additive codes. The cyclic quantum codes correspond to a special case when the codes are self-orthogonal with respect to the symplectic bilinear form.

Up to this point the codes considered are block codes where codewords all have fixed length. That is no longer the case in Chapter **10**, coauthored by Julia Lieb, Raquel Pinto, and Joachim Rosenthal. This chapter provides a survey of *convolutional codes* stressing the connections to module theory and systems theory. Constructions of codes with maximal possible distance and distance profile are provided.

Chapter **11**, written by Elisa Gorla, provides a mathematical introduction to *rank-metric codes*, beginning with the definition of the rank metric and the corresponding codes, whose elements can be either vectors or matrices. This is followed by the definition of code equivalence and the notion of support for a codeword and for a code. This chapter treats some of the basic concepts in the mathematical theory of rank-metric codes: duality, weight

enumerators and the MacWilliams Identities, higher rank weights, MRD codes, optimal anticodes, and $q$-polymatroids associated to a rank-metric code.

The final two chapters of Part I deal with the important technique of linear programming and a related generalization to produce bounds. As described in Chapter **12**, coauthored by Peter Boyvalenkov and Danyo Danev, general *linear programming* methods imply universal bounds for codes and designs. The explanation is organized in the Levenshtein framework extended with recent developments on universal bounds for the energy of codes, including the concept of universal optimality. The exposition is done separately for codes in Hamming spaces and for spherical codes.

Linear programming bounds, initially developed by Delsarte, belong to the most powerful and flexible methods to obtain bounds for extremal problems in coding theory. In recent years, after the pioneering work of Schrijver, *semidefinite programming* bounds have been developed with two aims: to strengthen linear programming bounds and to find bounds for more general spaces. Chapter **13**, by Frank Vallentin, introduces semidefinite programming bounds with an emphasis on error-correcting codes and its relation to semidefinite programming hierarchies for difficult combinatorial optimization problems.

The next eight chapters make up Part II of the *Concise Encyclopedia of Coding Theory*, where the focus is on specific families of codes. The codes presented fall into two categories, with some overlap. Some of them are generalizations of classical codes from Part I. The rest have a direct connection to algebraic, geometric, or graph theoretic structures. They all are interesting theoretically and often possess properties useful for application.

There are many problems in coding theory which are equivalent to geometrical problems in Galois geometries. Certain formulations of some of the classical codes have direct connections to geometry. Chapter **14**, written by Leo Storme, describes a number of the many links between *coding theory and Galois geometries*, and shows how these two research areas influence and stimulate each other.

Chapter **15**, written by Alain Couvreur and Hugues Randriambololona, surveys the development of the theory of *algebraic geometry codes* since their discovery in the late 1970s. The authors summarize the major results on various problems such as asymptotic parameters, improved estimates on the minimum distance, and decoding algorithms. In addition, the chapter describes various modern applications of these codes such as public-key cryptography, algebraic complexity theory, multiparty computation, and distributed storage.

Very often the parameters of good/optimal linear codes can be realized by *group codes*, that is, ideals in a group algebra $\mathbb{F}G$ where $G$ is a finite group and $\mathbb{F}$ is a finite field. Such codes, the topic of Chapter **16** written by Wolfgang Willems, carry more algebraic structure than only linear codes, which leads to an easier analysis of the codes. In particular, the full machinery of representation theory of finite groups can be applied to prove interesting coding theoretical properties.

Chapter **17**, coauthored by Hai Q. Dinh and Sergio R. López-Permouth, discusses foundational and theoretical aspects of the role of finite rings as alphabets in coding theory, with a concentration on the class of *constacyclic codes* over finite commutative chain rings. The chapter surveys both the simple-root and repeated-root cases. Several directions in which the notion of constacyclicity has been extended are also presented.

The next three chapters focus on codes with few weights; such codes have applications delineated throughout this encyclopedia. As described in Chapter **18**, written by Minjia Shi, one important construction technique for few-weight codes is to use *trace codes*. For example the simplex code, a one-weight code, can be constructed as a trace code by using finite field extensions. In recent years, this technique has been refined by using ring extensions of a

finite field coupled with a linear Gray map. Moreover, these image codes can be applied to secret sharing schemes.

Codes with few weights often have an interesting geometric structure. Chapter **19**, written by Andries E. Brouwer, focuses specifically on *codes with exactly two nonzero weights*. The chapter discusses the relationship between two-weight linear codes, strongly regular graphs, and 2-character subsets of a projective space.

Functions in general and more specifically cryptographic functions, that is highly nonlinear functions (PN, APN, bent, AB, plateaued), have important applications in coding theory since they are used to construct optimal linear codes and linear codes useful for applications such as secret sharing, two-party computation, and storage. The ultimate goal of Chapter **20**, by Sihem Mesnager, is to provide an overview and insights into *linear codes* with good parameters that are *constructed from functions* and polynomials over finite fields using multiple approaches.

Chapter **21** by Christine A. Kelley, the concluding chapter of Part II, gives an overview of *graph-based codes* and iterative message-passing decoding algorithms for these codes. Some important classes of low-density parity check codes, such as finite geometry codes, expander codes, protograph codes, and spatially coupled codes are discussed. Moreover, analysis techniques of the decoding algorithm for both the finite length case and the asymptotic length case are summarized. While the area of codes over graphs is vast, a few other families such as repeat accumulate and turbo-like codes are briefly mentioned.

The final chapter of Part II provides a natural bridge to the applications in Part III as codes from graphs were designed to facilitate communication. The thirteen chapters of Part III examine several applications that fall into two categories, again with some overlap. Some of the applications present codes developed for specific uses; other applications use codes to produce other structures that themselves become the main application.

The first chapter in Part III is again a bridge between the previous and successive chapters as it has a distinct theoretical slant but its content is useful as well in applications. Chapter **22**, written by Marcelo Firer, gives an account of many *metrics* used in the context of coding theory, mainly for decoding purposes. The chapter tries to stress the role of some metric related invariants and aspects that are eclipsed at the usual setting of the Hamming metric.

Chapter **23**, written by Alfred Wassermann, examines *algorithms for computer construction* of "good" linear codes and methods to determine the minimum distance and weight enumerator of a linear code. For code construction the focus is on the geometric view: a linear code can be seen as a suitable set of points in a projective geometry. The search then reduces to an integer linear programming problem. The chapter explores how the search space can be much reduced by prescribing a group of symmetries and how to construct special code types such as self-orthogonal codes or LCD codes.

In Chapter **24** by Swastik Kopparty we will see some algorithmic ideas based on *polynomial interpolation* for decoding algebraic codes, applied to generalized Reed–Solomon and interleaved generalized Reed–Solomon codes. These ideas will power decoding algorithms that can decode algebraic codes beyond half the minimum distance.

The theory of pseudo-noise sequences has close connections with coding theory, cryptography, combinatorics, and discrete mathematics. Chapter **25**, coauthored by Tor Helleseth and Chunlei Li, gives a brief introduction of two kinds of *pseudo-noise sequences*, namely sequences with low correlation and shift register sequences with maximum periods, which are of particular interest in modern communication systems.

*Lattice coding* is presented in Chapter **26**, written by Frédérique Oggier, in the context of Gaussian and fading channels, where channel models are presented. Lattice constructions

from both quadratic fields and linear codes are described. Some variations of lattice coding are also discussed.

A bridge between classical coding theory and quantum error control was firmly put in place via the stabilizer formalism, allowing the capabilities of a quantum stabilizer code to be inferred from the properties of the corresponding classical codes. Well-researched tools and the wealth of results in classical coding theory often translate nicely to the design of good *quantum codes*, the subject of Chapter **27** by Martianus Frederic Ezerman. Research problems triggered by error-control issues in the quantum setup revive and expand studies on specific aspects of classical codes, which were previously overlooked or deemed not so interesting.

Chapter **28** on *space-time coding*, written by Frédérique Oggier, defines what space-time coding actually is and what are variations of space-time coding problems. The chapter also provides channel models and design criteria, together with several examples.

Chapter **29**, by Frank R. Kschischang, describes *error-correcting network codes* for packet networks employing random linear network coding. In such networks, packets sent into the network by the transmitter are regarded as a basis for a vector space over a finite field, and the network provides the receiver with random linear combinations of the transmitted vectors, possibly also combined with noise vectors. Unlike classical coding theory—where codes are collections of well-separated *vectors*, each of them a point of some ambient vector space—here codes are collections of well-separated *vector spaces*, each of them a subspace of some ambient vector space. The chapter provides appropriate coding metrics for such subspace codes, and describes various bounds and constructions, focusing particularly on the case of constant-dimension codes whose codewords all have the same dimension.

*Erasure codes* have attained a position of importance for many streaming and file download applications on the internet. Chapter **30**, written by Ian F. Blake, outlines the development of these codes from simple erasure correcting codes to the important Raptor codes. Various decoding algorithms for these codes are developed and illustrated.

Chapter **31**, with coauthors Vinayak Ramkumar, Myna Vajha, S. B. Balaji, M. Nikhil Krishnan, Birenjith Sasidharan, and P. Vijay Kumar, deals with the topic of designing reliable and efficient *codes for the storage and retrieval* of large quantities of data over storage devices that are prone to failure. Historically, the traditional objective has been one of ensuring reliability against data loss while minimizing storage overhead. More recently, a third concern has surfaced, namely, the need to efficiently recover from the failure of a single storage unit corresponding to recovery from the erasure of a single code symbol. The authors explain how coding theory has evolved to tackle this fresh challenge.

Polar codes are error-correcting codes that achieve the symmetric capacity of discrete input memoryless channels with a polynomial encoding and decoding complexity. Chapter **32**, coauthored by Noam Presman and Simon Litsyn, provides a general presentation of *polar codes* and their associated algorithms. At the same time, most of the examples in the chapter use the basic Arıkan's $(u + v, v)$ original construction due to its simplicity and wide applicability.

While one thinks of coding theory as the major tool to reveal correct information after errors in that information have been introduced, the final two chapters of this encyclopedia address the opposite problem: using coding theory as a tool to hide information. Chapter **33**, by Cunsheng Ding, first gives a brief introduction to *secret sharing schemes*, and then introduces two constructions of secret sharing schemes with linear codes. It also documents a construction of multisecret sharing schemes with linear codes. Basic results about these secret sharing schemes are presented in this chapter.

Chapter **34**, written by Philippe Gaborit and Jean-Christophe Deneuville, gives a general overview of basic tools used for *code-based cryptography*. The security of the main difficult problem for code-based cryptography, the Syndrome Decoding problem, is considered,

together with its quasi-cyclic variations. The current state-of-the-art for the cryptographic primitives of encryption, signature, and authentication is the main focus of the chapter.

The editors of the *Concise Encyclopedia of Coding Theory* thank the 48 other authors for sharing their expertise to make this project come to pass. Their cooperation and patience were invaluable to us. We also thank Gayle Imamura-Huffman for lending her transparent watercolor *Coded Information* and opaque watercolor *Linear Subspaces* for use on the front and back covers of this encyclopedia. Additionally we thank the editorial staff at CRC Press/Taylor and Francis Group: Sarfraz Khan, who helped us begin this project; Callum Fraser, who became the Mathematical Editor at CRC Press as the project progressed; and Robin Lloyd-Starkes, who is Project Editor. Also with CRC Press, we thank Mansi Kabra for handling permissions and copyrights and Kevin Craig who assisted with the cover design. We thank Meeta Singh, Senior Project Manager at KnowledgeWorks Global Ltd., and her team for production of this encyclopedia. And of course, we sincerely thank our families for their support and encouragement throughout this journey.

– W. Cary Huffman
– Jon-Lark Kim
– Patrick Solé

# *Contributors*

**S. B. Balaji**
Qualcomm
Bangalore, India
balaji.profess@gmail.com

**Jürgen Bierbrauer**
Professor Emeritus
Department of Mathematical Sciences
Michigan Technological University
Houghton, Michigan, USA
jbierbra@mtu.edu

**Ian F. Blake**
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, British Columbia, Canada
ifblake@ece.ubc.ca

**Stefka Bouyuklieva**
Faculty of Mathematics and Informatics
St. Cyril and St. Methodius University
Veliko Tarnovo, Bulgaria
stefka@ts.uni-vt.bg

**Peter Boyvalenkov**
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Sofia, Bulgaria
*and* Technical Faculty
South-Western University
Blagoevgrad, Bulgaria
peter@math.bas.bg

**Andries E. Brouwer**
Department of Mathematics
Eindhoven University of Technology
Eindhoven, Netherlands
Andries.Brouwer@cwi.nl

**Alain Couvreur**
Inria *and* LIX
École Polytechnique
Palaiseau, France
alain.couvreur@inria.fr

**Danyo Danev**
Department of Electrical Engineering and Department of Mathematics
Linköping University
Linköping, Sweden
danyo.danev@liu.se

**Jean-Christophe Deneuville**
Ecole Nationale de l'Aviation Civile
University of Toulouse, France
jean-christophe.deneuville@enac.fr

**Cunsheng Ding**
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
cding@ust.hk

**Hai Q. Dinh**
Department of Mathematical Sciences
Kent State University
Warren, Ohio, USA
hdinh@kent.edu

**Steven T. Dougherty**
Department of Mathematics
University of Scranton
Scranton, Pennsylvania, USA
prof.steven.dougherty@gmail.com

**Martianus Frederic Ezerman**
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
fredezerman@ntu.edu.sg

**Marcelo Firer**
Department of Mathematics
State University of Campinas (Unicamp)
Campinas, Brasil
mfirer@ime.unicamp.br

**Philippe Gaborit**
XLIM-MATHIS
University of Limoges, France
gaborit@unilim.fr

**Heide Gluesing-Luerssen**
Department of Mathematics
University of Kentucky
Lexington, Kentucky, USA
heide.gl@uky.edu

**Elisa Gorla**
Institut de Mathématiques
Université de Neuchâtel
Neuchâtel, Switzerland
elisa.gorla@unine.ch

**Cem Güneri**
Faculty of Engineering and Natural Sciences
Sabancı University
Istanbul, Turkey
guneri@sabanciuniv.edu

**Tor Helleseth**
Department of Informatics
University of Bergen
Bergen, Norway
Tor.Helleseth@uib.no

**W. Cary Huffman**
Department of Mathematics and Statistics
Loyola University
Chicago, Illinois, USA
whuffma@luc.edu

**Christine A. Kelley**
Department of Mathematics
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
ckelley2@unl.edu

**Jon-Lark Kim**
Department of Mathematics
Sogang University
Seoul, South Korea
jlkim@sogang.ac.kr

**Swastik Kopparty**
Department of Mathematics
Rutgers University
Piscataway, New Jersey, USA
swastik.kopparty@gmail.com

**M. Nikhil Krishnan**
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada
nikhilkrishnan.m@gmail.com

**Frank R. Kschischang**
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada
frank@ece.utoronto.ca

**P. Vijay Kumar**
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore, India
*and*
Ming Hsieh Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, California, USA
pvk@iisc.ac.in, vijayk@usc.edu

**Chunlei Li**
Department of Informatics
University of Bergen
Bergen, Norway
chunlei.li@uib.no

**Julia Lieb**
Institut für Mathematik
Universität Zürich
Zürich, Switzerland
julia.lieb@math.uzh.ch

**San Ling**
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
lingsan@ntu.edu.sg

**Simon Litsyn**
Department of Electrical Engineering-Systems
Tel Aviv University
Ramat Aviv, Israel
litsyn@tauex.tau.ac.il

**Sergio R. López-Permouth**
Department of Mathematics
Ohio University
Athens, Ohio, USA
lopez@ohio.edu

**Stefano Marcugini**
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
stefano.marcugini@unipg.it

**Sihem Mesnager**
Department of Mathematics
University of Paris VIII, 93526 Saint-Denis, France
*and*
Laboratoire de Géométrie, Analyse et Applications, UMR 7539, CNRS
University Sorbonne Paris Cité, 93430 Villetaneuse, France
*and*
Télécom Paris
91120 Palaiseau, France
smesnager@univ-paris8.fr

**Frédérique Oggier**
Division of Mathematical Sciences
Nanyang Technological University
Singapore
frederique@ntu.edu.sg

**Patric R. J. Östergård**
Department of Communications and Networking
Aalto University
Espoo, Finland

**Buket Özkaya**
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
buketozkaya@ntu.edu.sg

**Fernanda Pambianco**
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
fernanda.pambianco@unipg.it

**Raquel Pinto**
Departamento de Matemática
Universidade de Aveiro
Aveiro, Portugal
raquel@ua.pt

**Noam Presman**
Department of Electrical Engineering-Systems
Tel Aviv University
Ramat Aviv, Israel
presmann@gmail.com

**Vinayak Ramkumar**
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore, India
vinram93@gmail.com

**Hugues Randriambololona**
ANSSI – Laboratoire de Cryptographie
Paris, France
*and*
Télécom Paris
Palaiseau, France
hugues.randriam@ssi.gouv.fr

**Joachim Rosenthal**
Institut für Mathematik
Universität Zürich
Zürich, Switzerland
rosenthal@math.uzh.ch

**Birenjith Sasidharan**
Department of Electronics and Communication Engineering
Govt. Engineering College, Barton Hill
Trivandrum, India
birenjith@gmail.com

**Minjia Shi**
School of Mathematical Sciences
Anhui University
Hefei, 230601, China
smjwcl.good@163.com

**Patrick Solé**
Lab I2M
CNRS, Aix-Marseille Université, Centrale Marseille
13 009 Marseilles, France
patrick.sole@telecom-paris.fr

**Leo Storme**
Department of Mathematics: Analysis, Logic and Discrete Mathematics
Ghent University
9000 Gent, Belgium
Leo.Storme@ugent.be

**Vladimir D. Tonchev**
Department of Mathematical Sciences
Michigan Technological University
Houghton, Michigan, USA
tonchev@mtu.edu

**Myna Vajha**
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore, India
mynaramana@gmail.com

**Frank Vallentin**
Mathematisches Institut
Universität zu Köln
Köln, Germany
frank.vallentin@uni-koeln.de

**Alfred Wassermann**
Mathematisches Institut
Universität Bayreuth
95440 Bayreuth, Germany
alfred.wassermann@uni-bayreuth.de

**Wolfgang Willems**
Fakultät für Mathematik
Otto-von-Guericke Universität
Magdeburg, Germany
*and*
Departamento de Matemáticas y Estadística
Universidad del Norte
Barranquilla, Colombia
willems@ovgu.de

# Part I

# Coding Fundamentals

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

# Chapter 1

# Basics of Coding Theory

**W. Cary Huffman**

*Loyola University, Chicago*

**Jon-Lark Kim**

*Sogang University*

**Patrick Solé**

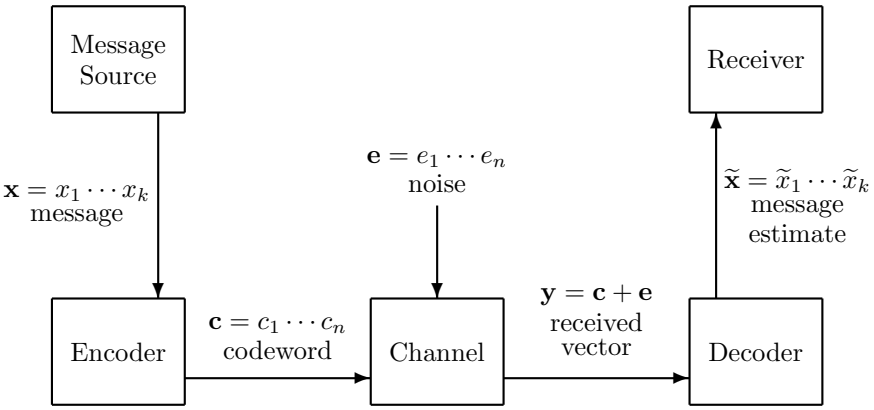
*CNRS, Aix-Marseille Université*



## 1.1 Introduction

Coding theory had it genesis in the late 1940s with the publication of works by Claude Shannon, Marcel Golay, and Richard Hamming. In 1948 Shannon published a landmark

**FIGURE 1.1**: A simple communication channel




paper *A mathematical theory of communication* [1661] which marked the beginning of both information theory and coding theory. Given a communication channel, over which information is transmitted and possibly corrupted, Shannon identified a number called the 'channel capacity' and proved that arbitrarily reliable communication is possible at any rate below the channel capacity. For example, when transmitting images of planets from deep space, it is impractical to retransmit the images that have been altered by noise during transmission. Shannon's Theorem guarantees that the data can be encoded before transmission so that the altered data can be decoded to the original, up to a specified degree of accuracy. Other examples of communication channels include wireless communication devices and storage systems such as DVDs or Blue-ray discs. In 1947 Hamming developed a code, now bearing his name, in an attempt to correct errors that arose in the Bell Telephone Laboratories' mechanical relay computer; his work was circulated through a series of memoranda at Bell Labs and eventually published in [895]. Both Shannon [1661] and Golay [820] published Hamming's code, with Golay generalizing it. Additionally, Golay presented two of the four codes that now bear his name. A monograph by T. M. Thompson [1801] traces the early development of coding theory.

A simple **communication channel** is illustrated in Figure 1.1. At the source a **message**, denoted $\mathbf{x} = x_1 \cdots x_k$ in the figure, is to be sent. If no modification is made to $\mathbf{x}$ and it is transmitted directly over the channel, any noise would distort $\mathbf{x}$ so that it could not be recovered. The basic idea of coding theory is to embellish the message by adding some redundancy so that hopefully the original message can be recovered after reception even if noise corrupts the embellished message during transmission. The redundancy is added by the encoder and the embellished message, called a **codeword** $\mathbf{c} = c_1 \cdots c_n$ in the figure, is sent over the channel where noise in the form of an **error vector** $\mathbf{e} = e_1 \cdots e_n$ distorts the codeword producing a received vector $\mathbf{y}$.[1] The received vector is then sent to be decoded where the errors are removed. The redundancy is then stripped off, and an estimate $\widetilde{\mathbf{x}}$ of the original message is produced. Hopefully $\widetilde{\mathbf{x}} = \mathbf{x}$. (There is a one-to-one correspondence

[1]Generally message and codeword symbols will come from a finite field $\mathbb{F}$ or a finite ring $R$. Messages will be 'vectors' in $\mathbb{F}^k$ (or $R^k$), and codewords will be 'vectors' in $\mathbb{F}^n$ (or $R^n$). If $\mathbf{c}$ entered the channel and $\mathbf{y}$ exited the channel, the difference $\mathbf{y} - \mathbf{c}$ is what we have termed the error vector $\mathbf{e}$ in Figure 1.1. While this is the normal scenario, other ambient spaces from which codes arise occur in this encyclopedia.

between codewords and messages. In many cases, the real interest is not in the message $\mathbf{x}$ but the codeword $\mathbf{c}$. With this point of view, the job of the decoder is to obtain an estimate $\widetilde{\mathbf{y}}$ from $\mathbf{y}$ and hope that $\widetilde{\mathbf{y}} = \mathbf{c}$.) For example in deep space communication, the message source is the satellite, the channel is outer space, the decoder is hardware at a ground station on Earth, and the receiver is the people or computer processing the information; of course, messages travel from Earth to the satellite as well. For a DVD or Blue-ray disc, the message source is the voice, music, video, or data to be placed on the disc, the channel is the disc itself, the decoder is the DVD or Blue-ray player, and the receiver is the listener or viewer.

Shannon's Theorem guarantees that the hope of successful recovery will be fulfilled a certain percentage of the time. With the right encoding based on the characteristics of the channel, this percentage can be made as high as desired, although not 100%. The proof of Shannon's Theorem is probabilistic and nonconstructive. No specific codes were produced in the proof that give the desired accuracy for a given channel. Shannon's Theorem only guarantees their existence. In essence, the goal of coding theory is to produce codes that fulfill the conditions of Shannon's Theorem and make reliable communication possible.

There are numerous texts, ranging from introductory to research-level books, on coding theory including (but certainly not limited to) [170, 209, 896, 1008, 1323, 1505, 1506, 1520, 1521, 1602, 1836]. There are two books, [169] edited by E. R. Berlekamp and [212] edited by I. F. Blake, in which early papers in the development of coding theory have been reprinted.

---

## 1.2 Finite Fields

Finite fields play an essential role in coding theory. The theory and construction of finite fields can be found, for example, in [1254] and [1408, Chapter 2]. Finite fields, as related specifically to codes, are described in [1008, 1323, 1602]. In this section we give a brief introduction.

**Definition 1.2.1** A **field** $\mathbb{F}$ is a nonempty set with two binary operations, denoted $+$ and $\cdot$, satisfying the following properties.

(a) For all $\alpha, \beta, \gamma \in \mathbb{F}$, $\alpha+\beta \in \mathbb{F}$, $\alpha\cdot\beta \in \mathbb{F}$, $\alpha+\beta = \beta+\alpha$, $\alpha\cdot\beta = \beta\cdot\alpha$, $\alpha+(\beta+\gamma) = (\alpha+\beta)+\gamma$, $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$, and $\alpha \cdot (\beta + \gamma) = \alpha \cdot \beta + \alpha \cdot \gamma$.

(b) $\mathbb{F}$ possesses an **additive identity** or **zero**, denoted 0, and a **multiplicative identity** or **unity**, denoted 1, such that $\alpha + 0 = \alpha$ and $\alpha \cdot 1 = \alpha$ for all $\alpha \in \mathbb{F}_q$.

(c) For all $\alpha \in \mathbb{F}$ and all $\beta \in \mathbb{F}$ with $\beta \neq 0$, there exists $\alpha' \in \mathbb{F}$, called the **additive inverse of** $\alpha$, and $\beta^* \in \mathbb{F}$, called the **multiplicative inverse of** $\beta$, such that $\alpha + \alpha' = 0$ and $\beta \cdot \beta^* = 1$.

The additive inverse of $\alpha$ will be denoted $-\alpha$, and the multiplicative inverse of $\beta$ will be denoted $\beta^{-1}$. Usually the multiplication operation will be suppressed; that is, $\alpha \cdot \beta$ will be denoted $\alpha\beta$. If $n$ is a positive integer and $\alpha \in \mathbb{F}$, $n\alpha = \alpha+\alpha+\cdots+\alpha$ ($n$ times), $\alpha^n = \alpha\alpha\cdots\alpha$ ($n$ times), and $\alpha^{-n} = \alpha^{-1}\alpha^{-1}\cdots\alpha^{-1}$ ($n$ times when $\alpha \neq 0$). Also $\alpha^0 = 1$ if $\alpha \neq 0$. The usual rules of exponentiation hold. If $\mathbb{F}$ is a finite set with $q$ elements, $\mathbb{F}$ is called a **finite field of order** $q$ and denoted $\mathbb{F}_q$.

**Example 1.2.2** Fields include the rational numbers $\mathbb{Q}$, the real numbers $\mathbb{R}$, and the complex numbers $\mathbb{C}$. Finite fields include $\mathbb{Z}_p$, the set of integers modulo $p$, where $p$ is a prime.

The following theorem gives some of the basic properties of finite fields.

**Theorem 1.2.3** *Let $\mathbb{F}_q$ be a finite field with $q$ elements. The following hold.*

(a) $\mathbb{F}_q$ *is unique up to isomorphism.*

(b) $q = p^m$ *for some prime $p$ and some positive integer $m$.*

(c) $\mathbb{F}_q$ *contains the subfield $\mathbb{F}_p = \mathbb{Z}_p$.*

(d) $\mathbb{F}_q$ *is a vector space over $\mathbb{F}_p$ of dimension $m$.*

(e) $p\alpha = 0$ *for all $\alpha \in \mathbb{F}_q$.*

(f) *If $\alpha, \beta \in \mathbb{F}_q$, $(\alpha + \beta)^p = \alpha^p + \beta^p$.*

(g) *There exists an element $\gamma \in \mathbb{F}_q$ with the following properties.*

(i) $\mathbb{F}_q = \{0, 1 = \gamma^0, \gamma, \ldots, \gamma^{q-2}\}$ *and* $\gamma^{q-1} = 1$,

(ii) $\{1 = \gamma^0, \gamma, \ldots, \gamma^{m-1}\}$ *is a basis of the vector space $\mathbb{F}_q$ over $\mathbb{F}_p$, and*

(iii) *there exist $a_0, a_1, \ldots, a_{m-1} \in \mathbb{F}_p$ such that*

$$\gamma^m = a_0 + a_1\gamma + \cdots + a_{m-1}\gamma^{m-1}. \tag{1.1}$$

(h) *For all $\alpha \in \mathbb{F}_q$, $\alpha^q = \alpha$.*

**Definition 1.2.4** In Theorem 1.2.3, $p$ is called the **characteristic** of $\mathbb{F}_q$. The element $\gamma$ is called a **primitive element** of $\mathbb{F}_q$.

**Remark 1.2.5** Using Theorem 1.2.3(f), the map $\sigma_p : \mathbb{F}_q \to \mathbb{F}_q$ given by $\sigma_p(\alpha) = \alpha^p$ is an automorphism of $\mathbb{F}_q$, called the **Frobenius automorphism of** $\mathbb{F}_q$. Once one primitive element $\gamma$ of $\mathbb{F}_q$ is found, the remaining primitive elements of $\mathbb{F}_q$ are precisely $\gamma^d$ where $\gcd(d, q-1) = 1$.

The key to constructing a finite field is to find a primitive element $\gamma$ in $\mathbb{F}_q$ and the equation (1.1). We do not describe this process here, but refer the reader to the texts mentioned at the beginning of the section. Assuming $\gamma$ is found and the equation (1.1) is known, we can construct addition and multiplication tables for $\mathbb{F}_q$. This is done by writing every element of $\mathbb{F}_q$ in two forms. The first form takes advantage of Theorem 1.2.3(g)(ii). Every element $\alpha \in \mathbb{F}_q$ is written uniquely in the form

$$\alpha = a_0\gamma^0 + a_1\gamma + a_2\gamma^2 + \cdots + a_{m-1}\gamma^{m-1} \quad \text{with } a_i \in \mathbb{F}_p = \mathbb{Z}_p \text{ for } 0 \le i \le m-1,$$

which we abbreviate $\alpha = a_0a_1a_2\cdots a_{m-1}$, a vector in $\mathbb{Z}_p^m$. Addition in $\mathbb{F}_q$ is accomplished by ordinary vector addition in $\mathbb{Z}_p^m$. To each $\alpha \in \mathbb{F}_q$, with $\alpha \neq 0$, we associate a second form: $\alpha = \gamma^i$ for some $i$ with $0 \le i \le q-2$. Multiplication is accomplished by $\gamma^i\gamma^j = \gamma^{i+j}$ where we use $\gamma^{q-1} = 1$ when appropriate. We illustrate this by constructing the field $\mathbb{F}_9$.

**Example 1.2.6** The field $\mathbb{F}_9$ has characteristic 3 and is a 2-dimensional vector space over $\mathbb{Z}_3$. One primitive element $\gamma$ of $\mathbb{F}_9$ satisfies $\gamma^2 = 1 + \gamma$. Table 1.1 gives the two forms of all elements. The zero element is $0\gamma^0 + 0\gamma = 00$; the unity element is $1 = 1\gamma^0 + 0\gamma = 10$. Now $\gamma = 0\gamma^0 + 1\gamma = 01$, $\gamma^2 = 1+\gamma = 1\gamma^0+1\gamma = 11$, $\gamma^3 = \gamma\gamma^2 = \gamma(1+\gamma) = \gamma+\gamma^2 = \gamma+(1+\gamma) = 1\gamma^0+2\gamma = 12$, and $\gamma^4 = \gamma\gamma^3 = \gamma(1+2\gamma) = \gamma+2\gamma^2 = \gamma+2(1+\gamma) = 2\gamma^0+0\gamma = 20$. Note $\gamma^4 = -1$. $\gamma^5, \gamma^6, \gamma^7$ are computed similarly. As an example, we compute $(\gamma^5-1+\gamma^6)/(\gamma^5+\gamma^3+1)$ as follows. First $\gamma^5-1+\gamma^6 = 02-10+22 = 11 = \gamma^2$, and $\gamma^5+\gamma^3+1 = 02+12+10 = 21 = \gamma^7$. So $(\gamma^5 - 1 + \gamma^6)/(\gamma^5 + \gamma^3 + 1) = \gamma^2/\gamma^7 = \gamma^{-5} = \gamma^3$ since $\gamma^8 = 1$.

Tables 1.2, 1.3, and 1.4 give addition and multiplication tables for $\mathbb{F}_4$, $\mathbb{F}_8$, and $\mathbb{F}_{16}$, respectively. These fields have characteristic 2. Notice that $\mathbb{F}_{16}$ contains the subfield $\mathbb{F}_4$ where $\omega = \rho^5$.

**TABLE 1.1**: $\mathbb{F}_9$ with primitive element $\gamma$ where $\gamma^2 = 1 + \gamma$ and $\gamma^8 = 1$

| vector | power of $\gamma$ | vector | power of $\gamma$ | vector | power of $\gamma$ |
|---|---|---|---|---|---|
| 00 | — | 11 | $\gamma^2$ | 02 | $\gamma^5$ |
| 10 | $\gamma^0 = 1$ | 12 | $\gamma^3$ | 22 | $\gamma^6$ |
| 01 | $\gamma$ | 20 | $\gamma^4 = -1$ | 21 | $\gamma^7$ |

**TABLE 1.2**: $\mathbb{F}_4$ with primitive element $\omega$ where $\omega^2 = 1 + \omega$ and $\omega^3 = 1$

| vector | power of $\omega$ | vector | power of $\omega$ | vector | power of $\omega$ | vector | power of $\omega$ |
|---|---|---|---|---|---|---|---|
| 00 | — | 10 | $\omega^0 = 1$ | 01 | $\omega$ | 11 | $\omega^2$ |

## 1.3 Codes

In this section we introduce the concept of codes over finite fields. We begin with some notation.

The **set of $n$-tuples with entries in $\mathbb{F}_q$** forms an $n$-dimensional vector space, denoted $\mathbb{F}_q^n = \{x_1x_2\cdots x_n \mid x_i \in \mathbb{F}_q,\ 1 \le i \le n\}$, under componentwise addition of $n$-tuples and componentwise multiplication of $n$-tuples by scalars in $\mathbb{F}_q$. The vectors in $\mathbb{F}_q^n$ will often be denoted using bold Roman characters $\mathbf{x} = x_1x_2\cdots x_n$. The vector $\mathbf{0} = 00\cdots0$ is the **zero vector** in $\mathbb{F}_q^n$.

For positive integers $m$ and $n$, $\mathbb{F}_q^{m\times n}$ denotes **the set of all $m \times n$ matrices with entries in $\mathbb{F}_q$**. The matrix in $\mathbb{F}_q^{m\times n}$ with all entries 0 is the **zero matrix** denoted $\mathbf{0}_{m\times n}$. The **identity matrix** of $\mathbb{F}_q^{n\times n}$ will be denoted $I_n$. If $A \in \mathbb{F}_q^{m\times n}$, $A^{\mathsf{T}} \in \mathbb{F}_q^{n\times m}$ will denote the **transpose** of $A$. If $\mathbf{x} \in \mathbb{F}_q^m$, $\mathbf{x}^{\mathsf{T}}$ will denote $\mathbf{x}$ as a column vector of length $m$, that is, an $m \times 1$ matrix. The column vector $\mathbf{0}^{\mathsf{T}}$ and the $m \times 1$ matrix $\mathbf{0}_{m\times1}$ are the same.

If $S$ is any finite set, its **order** or **size** is denoted $|S|$.

**Definition 1.3.1** A subset $\mathcal{C} \subseteq \mathbb{F}_q^n$ is called a **code of length $n$ over $\mathbb{F}_q$**; $\mathbb{F}_q$ is called the **alphabet** of $\mathcal{C}$, and $\mathbb{F}_q^n$ is the **ambient space** of $\mathcal{C}$. Codes over $\mathbb{F}_q$ are also called **$q$-ary codes**. If the alphabet is $\mathbb{F}_2$, $\mathcal{C}$ is **binary**. If the alphabet is $\mathbb{F}_3$, $\mathcal{C}$ is **ternary**. The vectors in $\mathcal{C}$ are the **codewords** of $\mathcal{C}$. If $\mathcal{C}$ has $M$ codewords (that is, $|\mathcal{C}| = M$) $\mathcal{C}$ is denoted an $(n, M)_q$ code, or, more simply, an $(n, M)$ code when the alphabet $\mathbb{F}_q$ is understood. If $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$, that is $\mathcal{C}$ is closed under vector addition and scalar multiplication, $\mathcal{C}$ is called a **linear code of length $n$ over $\mathbb{F}_q$**. If the dimension of the linear code $\mathcal{C}$ is $k$, $\mathcal{C}$ is denoted an $[n, k]_q$ code, or, more simply, an $[n, k]$ code. An $(n, M)_q$ code that is also linear is an $[n, k]_q$ code where $M = q^k$. An $(n, M)_q$ code may be referred to as an **unrestricted code**; a specific unrestricted code may be either linear or nonlinear. When referring to a code, expressions such as $(n, M)$, $(n, M)_q$, $[n, k]$, or $[n, k]_q$ are called the **parameters** of the code.

**Example 1.3.2** Let $\mathcal{C} = \{1100, 1010, 1001, 0110, 0101, 0011\} \subseteq \mathbb{F}_2^4$. Then $\mathcal{C}$ is a $(4, 6)_2$ binary nonlinear code. Let $\mathcal{C}_1 = \mathcal{C} \cup \{0000, 1111\}$. Then $\mathcal{C}_1$ is a $(4, 8)_2$ binary linear code. As $\mathcal{C}_1$ is a subspace of $\mathbb{F}_2^4$ of dimension 3, $\mathcal{C}_1$ is also a $[4, 3]_2$ code.

**Remark 1.3.3** Basic development of linear codes is found in papers by D. Slepian [1722, 1723, 1724]. In some chapters of this book, codes will be considered where the alphabet is

**TABLE 1.3**: $\mathbb{F}_8$ with primitive element $\delta$ where $\delta^3 = 1 + \delta$ and $\delta^7 = 1$

| vector | power of $\delta$ | vector | power of $\delta$ | vector | power of $\delta$ | vector | power of $\delta$ |
|---|---|---|---|---|---|---|---|
| 000 | — | 010 | $\delta$ | 110 | $\delta^3$ | 111 | $\delta^5$ |
| 100 | $\delta^0 = 1$ | 001 | $\delta^2$ | 011 | $\delta^4$ | 101 | $\delta^6$ |

**TABLE 1.4**: $\mathbb{F}_{16}$ with primitive element $\rho$ where $\rho^4 = 1 + \rho$ and $\rho^{15} = 1$

| vector | power of $\rho$ | vector | power of $\rho$ | vector | power of $\rho$ | vector | power of $\rho$ |
|---|---|---|---|---|---|---|---|
| 0000 | — | 0001 | $\rho^3$ | 1101 | $\rho^7$ | 0111 | $\rho^{11}$ |
| 1000 | $\rho^0 = 1$ | 1100 | $\rho^4$ | 1010 | $\rho^8$ | 1111 | $\rho^{12}$ |
| 0100 | $\rho$ | 0110 | $\rho^5$ | 0101 | $\rho^9$ | 1011 | $\rho^{13}$ |
| 0010 | $\rho^2$ | 0011 | $\rho^6$ | 1110 | $\rho^{10}$ | 1001 | $\rho^{14}$ |

not necessarily a field but rather a ring $R$. In these situations, the vector space $\mathbb{F}_q^n$ will be replaced by an $R$-module such as $R^n = \{x_1 x_2 \cdots x_n \mid x_i \in R,\ 1 \leq i \leq n\}$, and a code will be considered *linear* if it is an $R$-submodule of that $R$-module. See for example Chapters 6, 17, and 18.

## 1.4 Generator and Parity Check Matrices

When choosing between linear and nonlinear codes, the added algebraic structure of linear codes often makes them easier to describe and use. Generally, a linear code is defined by giving either a generator or a parity check matrix.

**Definition 1.4.1** Let $\mathcal{C}$ be an $[n, k]_q$ linear code. A **generator matrix** $G$ **for** $\mathcal{C}$ is any $G \in \mathbb{F}_q^{k \times n}$ whose row span is $\mathcal{C}$. Because any $k$-dimensional subspace of $\mathbb{F}_q^n$ is the kernel of some linear transformation from $\mathbb{F}_q^n$ onto $\mathbb{F}_q^{n-k}$, there exists $H \in \mathbb{F}_q^{(n-k)\times n}$, with independent rows, such that $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n \mid H\mathbf{c}^\mathsf{T} = \mathbf{0}^\mathsf{T}\}$. Such a matrix, of which there are generally many, is called a **parity check matrix of** $\mathcal{C}$.

**Example 1.4.2** Continuing with Example 1.3.2, there are several generator matrices for $\mathcal{C}_1$ including

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix},\ G_1' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix},\ \text{and } G_1'' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

In this case there is only one parity check matrix $H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$.

**Remark 1.4.3** Any matrix obtained by elementary row operations from a generator matrix for a code remains a generator matrix of that code.

**Remark 1.4.4** By Definition 1.4.1, the rows of $G$ form a basis of $\mathcal{C}$, and the rows of $H$ are independent. At times, the requirement may be relaxed so that the rows of $G$ are only required to span $\mathcal{C}$. Similarly, the requirement that the rows of $H$ be independent may be dropped as long as $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n \mid H\mathbf{c}^\mathsf{T} = \mathbf{0}^\mathsf{T}\}$ remains true.

**Theorem 1.4.5 ([1323, Chapter 1.1])** *Let $G \in \mathbb{F}_q^{k\times n}$ and $H \in \mathbb{F}_q^{(n-k)\times n}$ each have independent rows. Let $\mathcal{C}$ be an $[n,k]_q$ code. The following hold.*

(a) *If $G$, respectively $H$, is a generator, respectively parity check, matrix for $\mathcal{C}$, then $HG^\mathsf{T} = \mathbf{0}_{(n-k)\times k}$.*

(b) *If $HG^\mathsf{T} = \mathbf{0}_{(n-k)\times k}$, then $G$ is a generator matrix for $\mathcal{C}$ if and only if $H$ is a parity check matrix for $\mathcal{C}$.*

**Definition 1.4.6** Let $\mathcal{C}$ be an $[n,k]_q$ linear code with generator matrix $G \in \mathbb{F}_q^{k\times n}$. For any set of $k$ independent columns of $G$, the corresponding set of coordinates forms an **information set** for $\mathcal{C}$; the remaining $n-k$ coordinates form a **redundancy set** for $\mathcal{C}$. If $G$ has the form $G = \left[I_k \mid A\right]$, $G$ is in **standard form** in which case $\{1,2,\ldots,k\}$ is an information set with $\{k+1,k+2,\ldots,n\}$ the corresponding redundancy set.

**Theorem 1.4.7 ([1602, Chapter 2.3])** *If $G = \left[I_k \mid A\right]$ is a generator matrix of an $[n,k]_q$ code $\mathcal{C}$, then $H = \left[-A^\mathsf{T} \mid I_{n-k}\right]$ is a parity check matrix for $\mathcal{C}$.*

**Example 1.4.8** Continuing with Examples 1.3.2 and 1.4.2, the matrix $G_1$ is in standard form. Applying Theorem 1.4.7 to $G_1$, we get the parity check matrix $H_1$ of Example 1.4.2. The matrices $G_1'$ and $G_1''$ both row reduce to $G_1$; so all three are generator matrices of the same code, consistent with Remark 1.4.3. Any subset of $\{1,2,3,4\}$ of size 3 is an information set for $\mathcal{C}_1$. The fact that $HG_1^\mathsf{T} = HG_1'^\mathsf{T} = HG_1''^\mathsf{T} = \mathbf{0}_{1\times 3}$ is consistent with Theorem 1.4.5. Finally, let $\mathcal{C}_2 = \{0000, 1100, 0011, 1111\}$ be the $[4,2]_2$ linear subcode of $\mathcal{C}_1$. $\mathcal{C}_2$ does not have a generator matrix in standard form; the only information sets for $\mathcal{C}_2$ are $\{1,3\}$, $\{1,4\}$, $\{2,3\}$, and $\{2,4\}$.

**Example 1.4.9** Generator and parity check matrices for the $[7,4]_2$ binary linear Hamming code $\mathcal{H}_{3,2}$ are

$$G_{3,2} = \left[\begin{array}{cccc|ccc} 1&0&0&0&0&1&1\\ 0&1&0&0&1&0&1\\ 0&0&1&0&1&1&0\\ 0&0&0&1&1&1&1 \end{array}\right] \quad \text{and} \quad H_{3,2} = \left[\begin{array}{cccc|ccc} 0&1&1&1&1&0&0\\ 1&0&1&1&0&1&0\\ 1&1&0&1&0&0&1 \end{array}\right],$$

respectively. $G_{3,2}$ is in standard form. Two information sets for $\mathcal{H}_{3,2}$ are $\{1,2,3,4\}$ and $\{1,2,3,5\}$ with corresponding redundancy sets $\{5,6,7\}$ and $\{4,6,7\}$. The set $\{2,3,4,5\}$ is not an information set. More general Hamming codes $\mathcal{H}_{m,q}$ are defined in Section 1.10.

## 1.5 Orthogonality

There is a natural inner product on $\mathbb{F}_q^n$ that often proves useful in the study of codes.[2]

**Definition 1.5.1** The **ordinary inner product**, also called the **Euclidean inner product**, on $\mathbb{F}_q^n$ is defined by $\mathbf{x}\cdot\mathbf{y} = \sum_{i=1}^n x_i y_i$ where $\mathbf{x} = x_1x_2\cdots x_n$ and $\mathbf{y} = y_1y_2\cdots y_n$. Two vectors $\mathbf{x},\mathbf{y} \in \mathbb{F}_q^n$ are **orthogonal** if $\mathbf{x}\cdot\mathbf{y} = 0$. If $\mathcal{C}$ is an $[n,k]_q$ code,

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x}\cdot\mathbf{c} = 0 \text{ for all } \mathbf{c} \in \mathcal{C}\}$$

[2]There are other inner products used in coding theory. See for example Chapters 4, 5, 7, 11, and 13.

is the **orthogonal code** or **dual code** of $\mathcal{C}$. $\mathcal{C}$ is **self-orthogonal** if $\mathcal{C} \subseteq \mathcal{C}^\perp$ and **self-dual** if $\mathcal{C} = \mathcal{C}^\perp$.

**Theorem 1.5.2 ([1323, Chapter 1.8])** *Let $\mathcal{C}$ be an $[n,k]_q$ code with generator and parity check matrices $G$ and $H$, respectively. Then $\mathcal{C}^\perp$ is an $[n, n-k]_q$ code with generator and parity check matrices $H$ and $G$, respectively. Additionally $(\mathcal{C}^\perp)^\perp = \mathcal{C}$. Furthermore $\mathcal{C}$ is self-dual if and only if $\mathcal{C}$ is self-orthogonal and $k = \frac{n}{2}$.*

**Example 1.5.3** $\mathcal{C}_2$ from Example 1.4.8 is a $[4,2]_2$ self-dual code with generator and parity check matrices both equal to

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

The dual of the Hamming $[7,4]_2$ code in Example 1.4.9 is a $[7,3]_2$ code $\mathcal{H}_{3,2}^\perp$. $H_{3,2}$ is a generator matrix of $\mathcal{H}_{3,2}^\perp$. As every row of $H_{3,2}$ is orthogonal to itself and every other row of $H_{3,2}$, $\mathcal{H}_{3,2}^\perp$ is self-orthogonal. As $\mathcal{H}_{3,2}^\perp$ has dimension 3 and $(\mathcal{H}_{3,2}^\perp)^\perp = \mathcal{H}_{3,2}$ has dimension 4, $\mathcal{H}_{3,2}^\perp$ is not self-dual.

## 1.6 Distance and Weight

The error-correcting capability of a code is keyed directly to the concepts of Hamming distance and Hamming weight.[3]

**Definition 1.6.1** The **(Hamming) distance** between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, denoted $\mathrm{d_H}(\mathbf{x}, \mathbf{y})$, is the number of coordinates in which $\mathbf{x}$ and $\mathbf{y}$ differ. The **(Hamming) weight** of $\mathbf{x} \in \mathbb{F}_q^n$, denoted $\mathrm{wt_H}(\mathbf{x})$, is the number of coordinates in which $\mathbf{x}$ is nonzero.

**Theorem 1.6.2 ([1008, Chapter 1.4])** *The following hold.*

(a) (nonnegativity) $\mathrm{d_H}(\mathbf{x}, \mathbf{y}) \geq 0$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$.

(b) $\mathrm{d_H}(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$.

(c) (symmetry) $\mathrm{d_H}(\mathbf{x}, \mathbf{y}) = \mathrm{d_H}(\mathbf{y}, \mathbf{x})$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$.

(d) (triangle inequality) $\mathrm{d_H}(\mathbf{x}, \mathbf{z}) \leq \mathrm{d_H}(\mathbf{x}, \mathbf{y}) + \mathrm{d_H}(\mathbf{y}, \mathbf{z})$ *for all* $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_q^n$.

(e) $\mathrm{d_H}(\mathbf{x}, \mathbf{y}) = \mathrm{wt_H}(\mathbf{x} - \mathbf{y})$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$.

(f) *If* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, *then*

$$\mathrm{wt_H}(\mathbf{x} + \mathbf{y}) = \mathrm{wt_H}(\mathbf{x}) + \mathrm{wt_H}(\mathbf{y}) - 2\mathrm{wt_H}(\mathbf{x} \star \mathbf{y})$$

*where $\mathbf{x} \star \mathbf{y}$ is the vector in $\mathbb{F}_2^n$ which has 1s precisely in those coordinates where both $\mathbf{x}$ and $\mathbf{y}$ have 1s.*

(g) *If* $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, *then* $\mathrm{wt_H}(\mathbf{x} \star \mathbf{y}) \equiv \mathbf{x} \cdot \mathbf{y} \pmod 2$. *In particular,* $\mathrm{wt_H}(\mathbf{x}) \equiv \mathbf{x} \cdot \mathbf{x} \pmod 2$.

[3] There are other notions of distance and weight used in coding theory. See for example Chapters 6, 7, 10, 11, 17, 18, 22, and 29.

(h) *If* $\mathbf{x} \in \mathbb{F}_3^n$, *then* $\mathrm{wt_H}(\mathbf{x}) \equiv \mathbf{x} \cdot \mathbf{x} \pmod 3$.

**Remark 1.6.3** A distance function on a vector space that satisfies parts (a) through (d) of Theorem 1.6.2 is called a **metric**; thus $\mathrm{d_H}$ is termed the **Hamming metric**. Other metrics useful in coding theory are examined in Chapter 22.

**Definition 1.6.4** Let $\mathcal{C}$ be an $(n, M)_q$ code with $M > 1$. The **minimum (Hamming) distance** of $\mathcal{C}$ is the smallest distance between distinct codewords. If the minimum distance $d$ of $\mathcal{C}$ is known, $\mathcal{C}$ is denoted an $(n, M, d)_q$ code (or an $[n, k, d]_q$ code if $\mathcal{C}$ is linear of dimension $k$). The **(Hamming) distance distribution** or **inner distribution** of $\mathcal{C}$ is the list $B_0(\mathcal{C}), B_1(\mathcal{C}), \ldots, B_n(\mathcal{C})$ where, for $0 \le i \le n$,

$$B_i(\mathcal{C}) = \frac{1}{M} \sum_{\mathbf{c} \in \mathcal{C}} \big|\{\mathbf{v} \in \mathcal{C} \mid \mathrm{d_H}(\mathbf{v}, \mathbf{c}) = i\}\big|.$$

The **minimum (Hamming) weight** of a nonzero code $\mathcal{C}$ is the smallest weight of nonzero codewords. The **(Hamming) weight distribution** of $\mathcal{C}$ is the list $A_0(\mathcal{C}), A_1(\mathcal{C}), \ldots, A_n(\mathcal{C})$ where, for $0 \le i \le n$, $A_i(\mathcal{C})$ is the number of codewords of weight $i$. If $\mathcal{C}$ is understood, the distance and weight distributions of $\mathcal{C}$ are denoted $B_0, B_1, \ldots, B_n$ and $A_0, A_1, \ldots, A_n$, respectively.

**Example 1.6.5** Let $\mathcal{C}$ be the $(4, 6)_2$ code in Example 1.3.2. Its distance distribution is $B_0(\mathcal{C}) = B_4(\mathcal{C}) = 1$, $B_2(\mathcal{C}) = 4$, $B_1(\mathcal{C}) = B_3(\mathcal{C}) = 0$, and its minimum distance is 2. In particular $\mathcal{C}$ is a $(4, 6, 2)_2$ code. The weight distribution of $\mathcal{C}$ is $A_2(\mathcal{C}) = 6$ with $A_i(\mathcal{C}) = 0$ otherwise; its minimum weight is also 2. Let $\mathcal{C}' = 1000 + \mathcal{C} = \{0100, 0010, 0001, 1110, 1101, 1011\}$. The distance distribution of $\mathcal{C}'$ agrees with the distance distribution of $\mathcal{C}$ making $\mathcal{C}'$ a $(4, 6, 2)_2$ code. However, the weight distribution of $\mathcal{C}'$ is $A_1(\mathcal{C}') = A_3(\mathcal{C}') = 3$ with $A_i(\mathcal{C}') = 0$ otherwise; the minimum weight of $\mathcal{C}'$ is 1.

**Theorem 1.6.6 ([1008, Chapter 1.4])** *Let* $\mathcal{C}$ *be an* $[n, k, d]_q$ *linear code with* $k > 0$. *The following hold.*

(a) *The minimum distance and minimum weight of* $\mathcal{C}$ *are the same.*

(b) $A_i(\mathcal{C}) = B_i(\mathcal{C})$ *for* $0 \le i \le n$.

(c) $\displaystyle\sum_{i=0}^{n} A_i(\mathcal{C}) = q^k$.

(d) $A_0(\mathcal{C}) = 1$ *and* $A_i(\mathcal{C}) = 0$ *for* $1 \le i < d$.

(e) *If* $q = 2$ *and* $\mathbf{1} = 11 \cdots 1 \in \mathcal{C}$, *then* $A_i(\mathcal{C}) = A_{n-i}(\mathcal{C})$ *for* $0 \le i \le n$.

(f) *If* $q = 2$ *and* $\mathcal{C}$ *is self-orthogonal, every codeword of* $\mathcal{C}$ *has even weight and* $\mathbf{1} \in \mathcal{C}^{\perp}$.

(g) *If* $q = 3$ *and* $\mathcal{C}$ *is self-orthogonal, every codeword of* $\mathcal{C}$ *has weight a multiple of* 3.

**Remark 1.6.7** Analogous to Theorem 1.6.6(c) and (d), if $\mathcal{C}$ is an $(n, M, d)_q$ code, then $\displaystyle\sum_{i=0}^{n} B_i(\mathcal{C}) = M$ with $B_0(\mathcal{C}) = 1$ and $B_i(\mathcal{C}) = 0$ for $1 \le i < d$.

Binary vectors possess an important relationship between weights and inner products. If $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$ and each have even weight, Theorem 1.6.2(f) implies $\mathbf{x} + \mathbf{y}$ also has even weight. If $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$ are orthogonal and each have weights a multiple of 4, Theorem 1.6.2(f) and (g) show that $\mathbf{x} + \mathbf{y}$ has weight a multiple of 4. This leads to the following definition for binary codes.

**Definition 1.6.8** Let $\mathcal{C}$ be a binary linear code. $\mathcal{C}$ is called **even** if all of its codewords have even weight. $\mathcal{C}$ is called **doubly-even** if all of its codewords have weights a multiple of 4. An even binary code that is not doubly-even is **singly-even**.

**Remark 1.6.9** By Theorem 1.6.6(e), self-orthogonal binary linear codes are even. The converse is not true; code $\mathcal{C}_1$ from Example 1.3.2 is even but not self-orthogonal. Doubly-even binary linear codes must be self-orthogonal by Theorem 1.6.2(f) and (g). There are self-orthogonal binary codes that are singly-even; code $\mathcal{C}_2$ from Examples 1.4.8 and 1.5.3 is singly-even and self-dual.

**Example 1.6.10** Let $\mathcal{H}_{3,2}$ be the $[7,4]_2$ binary Hamming code of Example 1.4.9. With $A_i = A_i(\mathcal{H}_{3,2})$, $A_0 = A_7 = 1$, $A_3 = A_4 = 7$, and $A_1 = A_2 = A_5 = A_6 = 0$, illustrating Theorem 1.6.6(d) and (e), and showing $\mathcal{H}_{3,2}$ is a $[7,4,3]_2$ code. The $[7,3]_2$ dual code $\mathcal{H}_{3,2}^{\perp}$ is self-orthogonal by Example 1.5.3 and hence even. Also by self-orthogonality, $\mathcal{H}_{3,2}^{\perp} \subseteq (\mathcal{H}_{3,2}^{\perp})^{\perp} = \mathcal{H}_{3,2}$; the weight distribution of $\mathcal{H}_{3,2}$ shows that the 8 codewords of weights 0 and 4 must be precisely the codewords of $\mathcal{H}_{3,2}^{\perp}$. In particular, $\mathcal{H}_{3,2}^{\perp}$ is a doubly-even $[7,3,4]_2$ code. $\mathcal{H}_{3,2}^{\perp}$ is called a **simplex** code, described further in Section 1.10.

The minimum weight of a linear code is determined by a parity check matrix for the code; see [1008, Corollary 1.4.14 and Theorem 1.4.15].

**Theorem 1.6.11** *A linear code has minimum weight $d$ if and only if its parity check matrix has a set of $d$ linearly dependent columns but no set of $d-1$ linearly dependent columns. Also, if $\mathcal{C}$ is an $[n,k,d]_q$ code, then every $n-d+1$ coordinate positions contain an information set; furthermore, $d$ is the largest number with this property.*

---

## 1.7 Puncturing, Extending, and Shortening Codes

There are several methods to obtain a longer or shorter code from a given code; while this can be done for both linear and nonlinear codes, we focus on linear ones. Two codes can be combined into a single code, for example as described in Section 1.11.

**Definition 1.7.1** Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code with generator matrix $G$ and parity check matrix $H$.

(a) For some $i$ with $1 \leq i \leq n$, let $\mathcal{C}^*$ be the codewords of $\mathcal{C}$ with the $i^{\text{th}}$ component deleted. The resulting code, called a **punctured code**, is an $[n-1,k^*,d^*]$ code. If $d > 1$, $k^* = k$, and $d^* = d$ unless $\mathcal{C}$ has a minimum weight codeword that is nonzero on coordinate $i$, in which case $d^* = d-1$. If $d = 1$, $k^* = k$ and $d^* = 1$ unless $\mathcal{C}$ has a weight 1 codeword that is nonzero on coordinate $i$, in which case $k^* = k-1$ and $d^* \geq 1$ as long as $\mathcal{C}^*$ is nonzero. A generator matrix for $\mathcal{C}^*$ is obtained from $G$ by deleting column $i$; $G^*$ will have dependent rows if $d^* = 1$ and $k^* = k-1$. Puncturing is often done on multiple coordinates in an analogous manner, one coordinate at a time.

(b) Define $\widehat{\mathcal{C}} = \left\{c_1c_2\cdots c_{n+1} \in \mathbb{F}_q^{n+1} \mid c_1c_2\cdots c_n \in \mathcal{C} \text{ where } \sum_{i=1}^{n+1} c_i = 0\right\}$, called the **extended code**. This is an $[n+1,k,\widehat{d}]_q$ code where $\widehat{d} = d$ or $d+1$. A generator

matrix $\widehat{G}$ for $\widehat{\mathcal{C}}$ is obtained by adding a column on the right of $G$ so that every row sum in this $k \times (n+1)$ matrix is 0. A parity check matrix $\widehat{H}$ for $\widehat{\mathcal{C}}$ is

$$\widehat{H} = \left[\begin{array}{ccc|c} 1 & \cdots & 1 & 1 \\ \hline & & & 0 \\ & H & & \vdots \\ & & & 0 \end{array}\right].$$

(c) Let $S$ be any set of $s$ coordinates. Let $\mathcal{C}(S)$ be all codewords in $\mathcal{C}$ that are zero on $S$. Puncturing $\mathcal{C}(S)$ on $S$ results in the $[n-s, k_S, d_S]_q$ **shortened code** $\mathcal{C}_S$ where $d_S \geq d$. If $\mathcal{C}^\perp$ has minimum weight $d^\perp$ and $s < d^\perp$, then $k_S = k - s$.

**Example 1.7.2** Let $\mathcal{H}_{3,2}$ be the $[7,4,3]_2$ binary Hamming code of Examples 1.4.9 and 1.6.10. Extending this code, we obtain $\widehat{\mathcal{H}}_{3,2}$ with generator and parity check matrices

$$\widehat{G}_{3,2} = \left[\begin{array}{cccc|cccc} 1&0&0&0&0&1&1&1 \\ 0&1&0&0&1&0&1&1 \\ 0&0&1&0&1&1&0&1 \\ 0&0&0&1&1&1&1&0 \end{array}\right] \text{ and } \widehat{H}_{3,2} = \left[\begin{array}{cccc|cccc} 1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&0&0&0 \\ 1&0&1&1&0&1&0&0 \\ 1&1&0&1&0&0&1&0 \end{array}\right],$$

respectively. Given the weight distribution of $\mathcal{H}_{3,2}$ found in Example 1.6.10, the weight distribution of $\widehat{\mathcal{H}}_{3,2}$ must be $A_0(\widehat{\mathcal{H}}_{3,2}) = A_8(\widehat{\mathcal{H}}_{3,2}) = 1$, $A_4(\widehat{\mathcal{H}}_{3,2}) = 14$, and $A_i(\widehat{\mathcal{H}}_{3,2}) = 0$ otherwise, implying $\widehat{\mathcal{H}}_{3,2}$ is doubly-even and self-dual; see Remark 1.6.9. Certainly if $\widehat{\mathcal{H}}_{3,2}$ is punctured on its right-most coordinate, the resulting code is $\mathcal{H}_{3,2}$.

There is a relationship between punctured and shortened codes via dual codes.

**Remark 1.7.3** If $\mathcal{C}$ is a linear code over $\mathbb{F}_q$ and $S$ a set of coordinates, then $(\mathcal{C}^\perp)_S = (\mathcal{C}^S)^\perp$ and $(\mathcal{C}^\perp)^S = (\mathcal{C}_S)^\perp$ where $\mathcal{C}^S$ and $(\mathcal{C}^\perp)^S$ are $\mathcal{C}$ and $\mathcal{C}^\perp$ punctured on $S$; see [1008, Theorem 1.5.7].

## 1.8 Equivalence and Automorphisms

Two vector spaces over $\mathbb{F}_q$ are considered the *same* (that is, isomorphic) if there is a nonsingular linear transformation from one to the other. For linear codes to be considered the *same*, we want these linear transformations to also preserve weights of codewords. In Theorem 1.8.6, we will see that these weight preserving linear transformations are directly related to monomial matrices. This leads to two different concepts of code equivalence for linear codes.

**Definition 1.8.1** If $P \in \mathbb{F}_q^{n\times n}$ has exactly one 1 in each row and column and 0 elsewhere, $P$ is a **permutation matrix**. If $M \in \mathbb{F}_q^{n\times n}$ has exactly one nonzero entry in each row and column, $M$ is a **monomial matrix**. If $\mathcal{C}$ is a code over $\mathbb{F}_q$ of length $n$ and $A \in \mathbb{F}_q^{n\times n}$, then $\mathcal{C}A = \{\mathbf{c}A \mid \mathbf{c} \in \mathcal{C}\}$. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be linear codes over $\mathbb{F}_q$ of length $n$. $\mathcal{C}_1$ is **permutation equivalent** to $\mathcal{C}_2$ provided $\mathcal{C}_2 = \mathcal{C}_1 P$ for some permutation matrix $P \in \mathbb{F}_q^{n\times n}$. $\mathcal{C}_1$ is **monomially equivalent** to $\mathcal{C}_2$ provided $\mathcal{C}_2 = \mathcal{C}_1 M$ for some monomial matrix $M \in \mathbb{F}_q^{n\times n}$.

**Remark 1.8.2** Applying a permutation matrix to a code simply permutes the coordinates; applying a monomial matrix permutes and re-scales coordinates. Applying either a permutation or monomial matrix to a vector does not change its weight. Also applying either a permutation or monomial matrix to two vectors does not change the distance between these two vectors. There is a third more general concept of equivalence, involving semi-linear transformations, where two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ over $\mathbb{F}_q$ are **equivalent** provided one can be obtained from the other by permuting and re-scaling coordinates and then applying an automorphism of the field $\mathbb{F}_q$. Note that applying such maps to a vector or to a pair of vectors preserves the weight of the vector and the distance between the two vectors, respectively; see [1008, Section 1.7] for further discussion of this type of equivalence. There are other concepts of equivalence that arise when the code may not be linear but has some specific algebraic structure (e.g., additive codes over $\mathbb{F}_q$ that are closed under vector addition but not necessarily closed under scalar multiplication). The common theme when defining equivalence of such codes is to use a set of maps which preserve distance between the two vectors, which preserve the algebraic structure under consideration, and which form a group under composition of these maps. We will follow this theme when we define equivalence of unrestricted codes at the end of this section.

**Remark 1.8.3** Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be linear codes over $\mathbb{F}_q$ of length $n$. Define $\mathcal{C}_1 \sim_P \mathcal{C}_2$ to mean $\mathcal{C}_1$ is permutation equivalent to $\mathcal{C}_2$; similarly define $\mathcal{C}_1 \sim_M \mathcal{C}_2$ to mean $\mathcal{C}_1$ is monomially equivalent to $\mathcal{C}_2$. Then both $\sim_P$ and $\sim_M$ are equivalence relations on the set of all linear codes over $\mathbb{F}_q$ of length $n$; that is, both are reflexive, symmetric, and transitive. If $q = 2$, the concepts of permutation and monomial equivalence are the same; if $q > 2$, they may not be. Furthermore, two permutation or monomially equivalent codes have the same size, weight and distance distributions, and minimum weight and distance. If two linear codes are permutation equivalent and one code is self-orthogonal, so is the other; this may not be true of two monomially equivalent codes.

Row reducing a generator matrix of a linear code to reduced echelon form and then permuting columns yields the following result.

**Theorem 1.8.4** *Let $\mathcal{C}$ be a linear $[n, k, d]_q$ code with $k \geq 1$. There is a code permutation equivalent to $\mathcal{C}$ with a generator matrix in standard form.*

**Example 1.8.5** Let $\mathcal{C}$ be an $[8, 4, 4]_2$ binary linear code. By Theorem 1.8.4, $\mathcal{C}$ is permutation equivalent to a code with generator matrix $G = \left[ I_4 \mid A \right]$. A straightforward argument using minimum weight 4 shows that columns of $A$ can be permuted so that the resulting generator matrix is $\widehat{G}_3$ from Example 1.7.2. This verifies that $\mathcal{C}$ is permutation equivalent to $\widehat{\mathcal{H}}_{3,2}$.

The following is a generalization of a result of MacWilliams [1318]; see also [229, 1876]. This result motivated Definition 1.8.1.

**Theorem 1.8.6 (MacWilliams Extension)** *There is a weight preserving linear transformation between equal length linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ over $\mathbb{F}_q$ if and only if $\mathcal{C}_1$ and $\mathcal{C}_2$ are monomially equivalent. Furthermore, the linear transformation agrees with the associated monomial transformation on every codeword in $\mathcal{C}_1$.*

**Definition 1.8.7** Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$ of length $n$. If $\mathcal{C}P = \mathcal{C}$ for some permutation matrix $P \in \mathbb{F}_q^{n \times n}$, then $P$ is a **permutation automorphism** of $\mathcal{C}$; the set of all permutation automorphisms of $\mathcal{C}$ is a group under matrix multiplication, denoted $\mathrm{PAut}(\mathcal{C})$. Similarly, if $\mathcal{C}M = \mathcal{C}$ for some monomial matrix $M \in \mathbb{F}_q^{n \times n}$, then $M$ is a **monomial automorphism** of $\mathcal{C}$; the set of all monomial automorphisms of $\mathcal{C}$ is a matrix group, denoted $\mathrm{MAut}(\mathcal{C})$. Clearly $\mathrm{PAut}(\mathcal{C}) \subseteq \mathrm{MAut}(\mathcal{C})$.

We now consider when two unrestricted codes are equivalent. It should be noted that, in this definition, a linear code may end up being equivalent to a nonlinear code. See Chapter 3 for more on this general equivalence.

**Definition 1.8.8** Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be unrestricted codes of length $n$ over $\mathbb{F}_q$ of the same size. Then $\mathcal{C}_1$ is **equivalent** to $\mathcal{C}_2$ provided the codewords of $\mathcal{C}_2$ are the images under a map of the codewords of $\mathcal{C}_1$ where the map is a permutation of coordinates together with $n$ permutations of the alphabet $\mathbb{F}_q$, independently within each coordinate.[4]

## 1.9 Bounds on Codes

In this section we present seven bounds relating the length, dimension or number of codewords, and minimum distance of an unrestricted code. The first five are considered upper bounds on the code size given length, minimum distance, and field size. By this, we mean that there does not exist a code of size bigger than the upper bound with the specified length, minimum distance, and field size. The last two are lower bounds on the size of a linear code. This means that a linear code can be constructed with the given length and minimum distance over the specified field having size equalling or exceeding the lower bound. We also give asymptotic versions of these bounds. Some of these bounds will be described using $A_q(n, d)$ and $B_q(n, d)$, which we now define.

**Definition 1.9.1** For positive integers $n$ and $d$, $A_q(n, d)$ is the **largest number of codewords in an** $(n, M, d)_q$ **code**, linear or nonlinear. $B_q(n, d)$ is the **largest number of codewords in a** $[n, k, d]_q$ **linear code**. An $(n, M, d)_q$ code is **optimal** provided $M = A_q(n, d)$; an $[n, k, d]_q$ linear code is **optimal** if $q^k = B_q(n, d)$. The concept of 'optimal' can also be used in other contexts. Given $n$ and $d$, $k_q(n, d)$ denotes the largest dimension of a linear code over $\mathbb{F}_q$ of length $n$ and minimum weight $d$; an $[n, k_q(n, d), d]_q$ code could be called 'optimal in dimension'. Notice that $k_q(n, d) = \log_q B_q(n, d)$. Similarly, $d_q(n, k)$ denotes the largest minimum distance of a linear code over $\mathbb{F}_q$ of length $n$ and dimension $k$; an $[n, k, d_q(n, k)]_q$ may be called 'optimal in distance'. Analogously, $n_q(k, d)$ denotes the smallest length of a linear code over $\mathbb{F}_q$ of dimension $k$ and minimum weight $d$; an $[n_q(k, d), k, d]_q$ code might be called 'optimal in length'.[5]

Clearly $B_q(n, d) \leq A_q(n, d)$. On-line tables relating parameters of various types of codes are maintained by M. Grassl [845].

The following basic properties of $A_q(n, d)$ and $B_q(n, d)$ are easily derived; see [1008, Chapter 2.1].

**Theorem 1.9.2** *The following hold for* $1 \leq d \leq n$.

(a) $B_q(n, d) \leq A_q(n, d)$.

(b) $B_q(n, n) = A_q(n, n) = q$ *and* $B_q(n, 1) = A_q(n, 1) = q^n$.

[4] In a more general setting, unrestricted codes do not have to have $\mathbb{F}_q$ as an alphabet. If $\mathfrak{A}$ is the alphabet, the permutations within each coordinate are permutations of $\mathfrak{A}$.

[5] Further restrictions might be placed on a family of codes when discussing optimality. For example, given $n$, a self-dual $[n, \frac{n}{2}, d]_q$ code over $\mathbb{F}_q$ with largest minimum weight $d$ is sometimes called an 'optimal $q$-ary self-dual code of length $n$'. Optimal codes are explored in chapters such as 2–5, 11, 12, 16, 18, 20, and 23.

(c) $B_q(n,d) \le qB_q(n-1,d)$ *and* $A_q(n,d) \le qA_q(n-1,d)$ *when* $1 \le d < n$.

(d) $B_q(n,d) \le B_q(n-1,d-1)$ *and* $A_q(n,d) \le A_q(n-1,d-1)$.

(e) *If* $d$ *is even,* $B_2(n,d) = B_2(n-1,d-1)$ *and* $A_2(n,d) = A_2(n-1,d-1)$.

(f) *If* $d$ *is even and* $M = A_2(n,d)$*, then there is an* $(n,M,d)_2$ *code such that all codewords have even weight and the distance between all pairs of codewords is also even.*

### 1.9.1 The Sphere Packing Bound

The Sphere Packing Bound, also called the Hamming Bound, is based on packing $\mathbb{F}_q^n$ with non-overlapping spheres.

**Definition 1.9.3** The **sphere of radius** $r$ **centered at** $\mathbf{u} \in \mathbb{F}_q^n$ is the set $S_{q,n,r}(\mathbf{u}) = \{\mathbf{v} \in \mathbb{F}_q^n \mid \mathrm{d_H}(\mathbf{u},\mathbf{v}) \le r\}$ of all vectors in $\mathbb{F}_q^n$ whose distance from $\mathbf{u}$ is at most $r$.

We need the size of a sphere, which requires use of binomial coefficients.

**Definition 1.9.4** For $a, b$ integers with $0 \le b \le a$, $\binom{a}{b}$ is the number of $b$-element subsets in an $a$-element set. $\binom{a}{b} = \dfrac{a!}{b!(a-b)!}$ and is called a **binomial coefficient**.

The next result is the basis of the Sphere Packing Bound; part (a) is a direct count and part (b) follows from the triangle inequality of Theorem 1.6.2.

**Theorem 1.9.5** *The following hold.*

(a) *For* $\mathbf{u} \in \mathbb{F}_q^n$, $|S_{q,n,r}(\mathbf{u})| = \displaystyle\sum_{i=0}^{r} \binom{n}{i}(q-1)^i$.

(b) *If* $\mathcal{C}$ *is an* $(n,M,d)_q$ *code and* $t = \left\lfloor \frac{d-1}{2} \right\rfloor$*, then spheres of radius* $t$ *centered at distinct codewords are disjoint.*

**Theorem 1.9.6 (Sphere Packing (or Hamming) Bound)** *Let* $d \ge 1$*. If* $t = \left\lfloor \frac{d-1}{2} \right\rfloor$*, then*

$$B_q(n,d) \le A_q(n,d) \le \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}.$$

**Proof:** Let $\mathcal{C}$ be an $(n,M,d)_q$ code. By Theorem 1.9.5, the spheres of radius $t$ centered at distinct codewords are disjoint, and each such sphere has $\alpha = \displaystyle\sum_{i=0}^{t} \binom{n}{i}(q-1)^i$ total vectors. Thus $M\alpha$ cannot exceed the number $q^n$ of vectors in $\mathbb{F}_q^n$. The result is now clear. □

**Remark 1.9.7** The Sphere Packing Bound is an upper bound on the size of a code given its length and minimum distance. Additionally the Sphere Packing Bound produces an upper bound on the minimum distance $d$ of an $(n,M)_q$ code in the following sense. Given $n$, $M$, and $q$, compute the smallest positive integer $s$ with $M > \dfrac{q^n}{\sum_{i=0}^{s} \binom{n}{i}(q-1)^i}$; for an $(n,M,d)_q$ code to exist, $d < 2s-1$.

**Definition 1.9.8** If $\mathcal{C}$ is an $(n, M, d)_q$ code with $M = \dfrac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}$ (that is, equality holds in the Sphere Packing Bound), $\mathcal{C}$ is called a **perfect code**. Perfect codes are precisely those $(n, M, d)_q$ codes where the disjoint spheres of radius $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ centered at all codewords fill the entire space $\mathbb{F}_q^n$. Perfect codes are discussed in Sections 3.3.1 and 5.3.

**Example 1.9.9** The code $\mathcal{H}_{3,2}$ of Examples 1.4.9 and 1.6.10 is a $[7, 4, 3]_2$ code. So in this case $t = \left\lfloor \frac{3-1}{2} \right\rfloor = 1$ and $\frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i} = \frac{2^7}{1+7} = 2^4$ yielding equality in the Sphere Packing Bound. So $\mathcal{H}_{3,2}$ is perfect.

### 1.9.2 The Singleton Bound

The Singleton Bound was formulated in [1717]. As with the Sphere Packing Bound, the Singleton Bound is an upper bound on the size of a code.

**Theorem 1.9.10 (Singleton Bound)** *For $d \leq n$, $A_q(n, d) \leq q^{n-d+1}$. Furthermore, if an $[n, k, d]_q$ linear code exists, then $k \leq n - d + 1$; i.e., $k_q(n, d) \leq n - d + 1$.*

**Remark 1.9.11** In addition to providing an upper bound on code size, the Singleton Bound yields the upper bound $d \leq n - \log_q(M) + 1$ on the minimum distance of an $(n, M, d)_q$ code.

**Definition 1.9.12** A code for which equality holds in the Singleton Bound is called **maximum distance separable (MDS)**. No code of length $n$ and minimum distance $d$ has more codewords than an MDS code with parameters $n$ and $d$; equivalently, no code of length $n$ with $M$ codewords has a larger minimum distance than an MDS code with parameters $n$ and $M$. MDS codes are discussed in Chapters 3, 6, 8, 14, and 33.

The following theorem is proved using Theorem 1.6.11.

**Theorem 1.9.13** *$\mathcal{C}$ is an $[n, k, n-k+1]_q$ MDS code if and only if $\mathcal{C}^\perp$ is an $[n, n-k, k+1]_q$ MDS code.*

**Example 1.9.14** Let $\mathcal{H}_{2,3}$ be the $[4, 2]_3$ ternary linear code with generator matrix

$$G_{2,3} = \left[\begin{array}{cc|cc} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{array}\right].$$

Examining inner products of the rows of $G_{2,3}$, we see that $\mathcal{H}_{2,3}$ is self-orthogonal of dimension half its length; so it is self-dual. Using Theorem 1.6.2(h), $A_0(\mathcal{H}_{2,3}) = 1$, $A_3(\mathcal{H}_{2,3}) = 8$, and $A_i(\mathcal{H}_{2,3}) = 0$ otherwise. In particular $\mathcal{H}_{2,3}$ is a $[4, 2, 3]_3$ code and hence is MDS.

### 1.9.3 The Plotkin Bound

The Binary Plotkin Bound [1527] is an upper bound on the size of an unrestricted binary code of length $n$ and minimum distance $d$ provided $d$ is close enough to $n$.

**Theorem 1.9.15 (Binary Plotkin Bound)** *Let $2d > n$. Then*

$$A_2(n, d) \leq 2 \left\lfloor \frac{d}{2d - n} \right\rfloor.$$

This result is generalized in [230] to unrestricted codes over $\mathbb{F}_q$.

**Theorem 1.9.16 (Generalized Plotkin Bound)** *If an* $(n, M, d)_q$ *code exists, then*

$$M(M-1)d \leq 2n \sum_{i=0}^{q-2} \sum_{j=i+1}^{q-1} M_i M_j$$

*where* $M_i = \left\lfloor \dfrac{M+i}{q} \right\rfloor$.

**Example 1.9.17** The Sphere Packing Bound yields $A_2(17,9) \leq \frac{131\,072}{3\,214}$ and $A_2(18,10) \leq \frac{262\,144}{4\,048}$; so $A_2(17,9) \leq 40$ and $A_2(18,10) \leq 64$. The Singleton Bound produces $A_2(17,9) \leq 512$ and $A_2(18,10) \leq 512$. The Binary Plotkin Bound gives $A_2(17,9) \leq 18$ and $A_2(18,10) \leq 10$. Using Theorem 1.9.2(e), the Plotkin Bound is best with $A_2(18,10) = A_2(17,9) \leq 10$. According to [845], there is a $(18,10,10)_2$ code implying $A_2(18,10) = A_2(17,9) = 10$.

### 1.9.4 The Griesmer Bound

The Griesmer Bound [855] is a lower bound on the length of a linear code given its dimension and minimum weight.

**Theorem 1.9.18 (Griesmer Bound)** *Let* $\mathcal{C}$ *be an* $[n,k,d]_q$ *linear code with* $k \geq 1$. *Then*

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

**Remark 1.9.19** One can interpret the Griesmer Bound as an upper bound on the code size given its length and minimum weight. Specifically, $B_q(n,d) \leq q^k$ where $k$ is the largest positive integer such that $n \geq \sum_{i=0}^{k-1} \lceil \frac{d}{q^i} \rceil$. This bound can also be interpreted as a lower bound on the length of a linear code of given dimension and minimum weight; that is, $n_q(k,d) \geq \sum_{i=0}^{k-1} \lceil \frac{d}{q^i} \rceil$. Finally, the Griesmer Bound can be understood as an upper bound on the minimum weight given the code length and dimension; given $n$ and $k$, $d_q(n,k)$ is at most the largest $d$ for which the bound holds.

**Example 1.9.20** Suppose we wish to find the smallest code length $n$ such that an $[n,4,3]_2$ code can exist. By the Griesmer Bound $n \geq \lceil \frac{3}{1} \rceil + \lceil \frac{3}{2} \rceil + \lceil \frac{3}{4} \rceil + \lceil \frac{3}{8} \rceil = 3+2+1+1 = 7$. Note that equality in this bound is attained by the $[7,4,3]_2$ code $\mathcal{H}_{3,2}$ of Examples 1.4.9 and 1.6.10.

### 1.9.5 The Linear Programming Bound

The Linear Programming Bound is a result of the work of P. Delsarte in [517, 519, 521]. This is generally the most powerful of the bounds but does require setting up and solving a linear program involving Krawtchouck polynomials.

**Definition 1.9.21** For $0 \leq k \leq n$, define the **Krawtchouck polynomial** $K_k^{(n,q)}(x)$ **of degree** $k$ to be

$$K_k^{(n,q)}(x) = \sum_{j=0}^{k} (-1)^j (q-1)^{k-j} \binom{x}{j} \binom{n-x}{k-j}.$$

An extensive presentation of properties of the Krawtchouck polynomials can be found in [1229, 1365] and in Section 12.1. A simple proof of the following result, known as the Delsarte–MacWilliams Inequalities, is found in [1885].

**Theorem 1.9.22 (Delsarte–MacWilliams Inequalities)** *Let $\mathcal{C}$ be an $(n, M, d)_q$ code with distance distribution $B_i(\mathcal{C})$ for $0 \leq i \leq n$. Then for $0 \leq k \leq n$*

$$\sum_{i=0}^{n} B_i(\mathcal{C}) K_k^{(n,q)}(i) \geq 0.$$

Let $\mathcal{C}$ be an $(n, M, d)_q$ code with distance distribution $B_i(\mathcal{C})$ for $0 \leq i \leq n$. By Remark 1.6.7, $M = \sum_{i=0}^{n} B_i(\mathcal{C})$, $B_0(\mathcal{C}) = 1$, and $B_i(\mathcal{C}) = 0$ for $1 \leq i \leq d-1$. Although $B_i(\mathcal{C})$ may not be an integer, $B_i(\mathcal{C}) \geq 0$. By the Delsarte–MacWilliams Inequalities, we also have $\sum_{i=0}^{n} B_i(\mathcal{C}) K_k^{(n,q)}(i) \geq 0$ for $0 \leq i \leq n$. As $K_0^{(n,q)}(i) = 1$, the $0^{\text{th}}$ Delsarte–MacWilliams Inequality is merely $\sum_{i=0}^{n} B_i(\mathcal{C}) \geq 0$, which is clearly already true. If $q = 2$, there are additional inequalities that hold. When $q = 2$, it is straightforward to show that $B_n(\mathcal{C}) \leq 1$. Furthermore when $q = 2$ and $d$ is even, we may also assume that $B_i(\mathcal{C}) = 0$ when $i$ is odd by Theorem 1.9.2(f). Properties of binomial coefficients show that $K_k^{(n,2)}(i) = (-1)^i K_{n-k}^{(n,2)}(i)$; thus the $k^{\text{th}}$ Delsarte–MacWilliams Inequality is the same as the $(n-k)^{\text{th}}$ Delsarte–MacWilliams Inequality because $B_i(\mathcal{C}) = 0$ when $i$ is odd. This discussion leads to the linear program that is set up to establish an upper bound on $A_q(n, d)$.

**Theorem 1.9.23 (Linear Programming Bound)** *The following hold.*

(a) *When $q \geq 2$, $A_q(n, d) \leq \max\{\sum_{i=0}^{n} B_i\}$ where the maximum is taken over all $B_i$ subject to the following conditions:*

  (i) *$B_0 = 1$ and $B_i = 0$ for $1 \leq i \leq d-1$,*

  (ii) *$B_i \geq 0$ for $d \leq i \leq n$, and*

  (iii) *$\sum_{i=0}^{n} B_i K_k^{(n,q)}(i) \geq 0$ for $1 \leq k \leq n$.*

(b) *When $d$ is even and $q = 2$, $A_2(n, d) \leq \max\{\sum_{i=0}^{n} B_i\}$ where the maximum is taken over all $B_i$ subject to the following conditions:*

  (i) *$B_0 = 1$ and $B_i = 0$ for $1 \leq i \leq d-1$ and all odd $i$,*

  (ii) *$B_i \geq 0$ for $d \leq i \leq n$ and $B_n \leq 1$, and*

  (iii) *$\sum_{i=0}^{n} B_i K_k^{(n,2)}(i) \geq 0$ for $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$.*

Sometimes additional constraints can be added to the linear program and reduce the size of $\max\{\sum_{i=0}^{n} B_i\}$. Linear Programming Bounds will be considered in more detail in Chapters 12 and 13.

### 1.9.6 The Gilbert Bound

The Gilbert Bound [806] is a lower bound on $B_q(n, d)$ and hence a lower bound on $A_q(n, d)$.

**Theorem 1.9.24 (Gilbert Bound)**

$$B_q(n, d) \geq \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i}.$$

### 1.9.7 The Varshamov Bound

The Varshamov Bound [1844] is similar to the Gilbert Bound; asymptotically they are the same as stated in Section 1.9.8.

**Theorem 1.9.25 (Varshamov Bound)**

$$B_q(n,d) \geq q^{n-\lceil \log_q(1+\sum_{i=0}^{d-2}\binom{n-1}{i}(q-1)^i)\rceil}.$$

### 1.9.8 Asymptotic Bounds

We now describe what happens to the bounds, excluding the Griesmer Bound, as the code length approaches infinity; these bounds are termed **asymptotic bounds**. We first need some terminology.

**Definition 1.9.26** The **information rate**, or simply **rate**, of an $(n, M, d)_q$ code is defined to be $\dfrac{\log_q M}{n}$. If the code is actually an $[n, k, d]_q$ linear code, its rate is $\dfrac{k}{n}$, measuring the number of information coordinates relative to the total number of coordinates. In either the linear or nonlinear case, the higher the rate, the higher the proportion of coordinates in a codeword that actually contain information rather than redundancy. The ratio $\dfrac{d}{n}$ is called the **relative distance** of the code; as we will see later, the relative distance is a measure of the error-correcting capability of the code relative to its length.

Each asymptotic bound will be either an upper or lower bound on the largest possible rate for a family of (possibly nonlinear) codes over $\mathbb{F}_q$ of lengths going to infinity with relative distances approaching $\delta$. The function, called the **asymptotic normalized rate function**, that determines this rate is

$$\alpha_q(\delta) = \limsup_{n\to\infty} \frac{\log_q A_q(n, \delta n)}{n}.$$

As the exact value of $\alpha_q(\delta)$ is unknown, we desire upper and lower bounds on this function. An upper bound would indicate that all families with relative distances approaching $\delta$ have rates, in the limit, at most this upper bound. A lower bound indicates that there exists a family of codes of lengths approaching infinity and relative distances approaching $\delta$ whose rates are at least this bound. Three of the bounds in the next theorem involve the entropy function.

**Definition 1.9.27** The **entropy function** is defined for $0 \leq x \leq r = 1 - q^{-1}$ by

$$H_q(x) = \begin{cases} 0 & \text{if } x = 0, \\ x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x) & \text{if } 0 < x \leq r. \end{cases}$$

Discussion and proofs of the asymptotic bounds can be found in [1008, 1323, 1505, 1836]. The **MRRW Bound**, named after the authors of [1365] who developed the bound, is the Asymptotic Linear Programming Bound. The MRRW Bound has been improved by M. Aaltonnen [2] in the case $q > 2$.

**Theorem 1.9.28 (Asymptotic Bounds)** *Let $q \geq 2$ and $r = 1 - q^{-1}$. The following hold.*

(a) (Asymptotic Sphere Packing) $\alpha_q(\delta) \leq 1 - H_q(\delta/2)$ *if* $0 < \delta \leq r$.

(b) (Asymptotic Singleton) $\alpha_q(\delta) \leq 1 - \delta$ *if* $0 \leq \delta \leq 1$.

(c) (Asymptotic Plotkin) $\alpha_q(\delta) = 0$ *if* $r \leq \delta \leq 1$ *and* $\alpha_q(\delta) \leq 1 - \frac{\delta}{r}$ *if* $0 \leq \delta \leq r$.

(d) (MRRW)

(i) (First MRRW)
$\alpha_q(\delta) \leq H_q\left(\frac{1}{q}\left(q - 1 - (q-2)\delta - 2\sqrt{(q-1)\delta(1-\delta)}\right)\right)$ *if* $0 < \delta < r$.

(ii) (Second MRRW) *Let* $g(x) = H_2((1 - \sqrt{1-x})/2)$.
$\alpha_2(\delta) \leq \min_{0 \leq u \leq 1-2\delta}\{1 + g(u^2) - g(u^2 + 2\delta u + 2\delta)\}$ *if* $0 < \delta < 1/2$.

(e) (Asymptotic Gilbert–Varshamov) $1 - H_q(\delta) \leq \alpha_q(\delta)$ *if* $0 < \delta \leq r$.

---

## 1.10 Hamming Codes

A binary code permutation equivalent to the code of Example 1.4.9 was discovered in 1947 by R. W. Hamming while working at Bell Telephone Laboratories. Because of patent considerations, his work was not published until 1950; see [895]. This Hamming code actually appeared earlier in C. E. Shannon's seminal paper [1661]. It was also generalized to codes over fields of prime order by M. J. E. Golay [820].

Given a positive integer $m$, if one takes an $m \times n$ binary matrix whose columns are nonzero and distinct, the binary code with this parity check matrix must have minimum weight at least 3 by Theorem 1.6.11. Binary Hamming codes $\mathcal{H}_{m,2}$ arise by choosing an $m \times n$ parity check matrix with the maximum number of columns possible that are distinct and nonzero.

**Definition 1.10.1** Let $m \geq 2$ be an integer and $n = 2^m - 1$. Let $H_{m,2}$ be an $m \times n$ matrix whose columns are all $2^m - 1$ distinct nonzero binary $m$-tuples. A code with this parity check matrix is called a **binary Hamming code**. Changing the column order of $H_{m,2}$ produces a set of pairwise permutation equivalent codes. Any code in this list is denoted $\mathcal{H}_{m,2}$ and is a $[2^m - 1, 2^m - 1 - m, 3]_2$ code.

The code $\mathcal{H}_{3,2}$ of Example 1.4.9 is indeed a binary Hamming code. These codes are generalized to Hamming codes $\mathcal{H}_{m,q}$ over $\mathbb{F}_q$, all with minimum weight 3 again from Theorem 1.6.11.

**Definition 1.10.2** Let $m \geq 2$ be an integer and $n = (q^m - 1)/(q-1)$. There are a total of $n$ 1-dimensional subspaces of $\mathbb{F}_q^m$. Let $H_{m,q}$ be an $m \times n$ matrix whose columns are all nonzero $m$-tuples with one column from each of the distinct 1-dimensional subspaces of $\mathbb{F}_q^m$. A code with this parity check matrix is called a **Hamming code over** $\mathbb{F}_q$. Re-scaling columns and/or changing column order of $H_{m,q}$ produces a set of pairwise monomially equivalent codes. Any code in this list is denoted $\mathcal{H}_{m,q}$ and is a $\left[(q^m-1)/(q-1), (q^m-1)/(q-1)-m, 3\right]_q$ code. The code $\mathcal{H}_{m,q}^{\perp}$ is called a **simplex code**.

**Example 1.10.3** The parity check matrix of the code in Example 1.9.14 is

$$\left[\begin{array}{cc|cc} -1 & -1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{array}\right].$$

This code satisfies the definition of a Hamming $[4, 2, 3]_3$ code, and so $\mathcal{H}_{2,3}$ is the appropriate labeling of this code.

The parameters of the Hamming codes in fact determine the code. That $\mathcal{H}_{m,q}$ is perfect follows by direct computation from Definition 1.9.8.

**Theorem 1.10.4** *The following hold.*

(a) *If $\mathcal{C}$ is a $[2^m-1, 2^m-1-m, 3]_2$ binary linear code, then $\mathcal{C}$ is permutation equivalent to $\mathcal{H}_{m,2}$.*

(b) *If $\mathcal{C}$ is a $\left[(q^m-1)/(q-1), (q^m-1)/(q-1)-m, 3\right]_q$ linear code, then $\mathcal{C}$ is monomially equivalent to $\mathcal{H}_{m,q}$.*

(c) *$\mathcal{H}_{m,q}$ is perfect.*

The weight distribution of $\mathcal{H}_{3,2}^{\perp}$ was given in Example 1.6.10. The following generalizes this; for a proof see [1008, Theorem 2.7.5].

**Theorem 1.10.5** *The nonzero codewords of the $\left[(q^m-1)/(q-1), m\right]_q$ simplex code over $\mathbb{F}_q$ all have weight $q^{m-1}$.*

## 1.11 Reed–Muller Codes

In 1954 the binary Reed–Muller codes were first constructed and examined by D. E. Muller [1409], and a majority logic decoding algorithm for them was described by I. S. Reed [1581]. The non-binary Reed–Muller codes, called generalized Reed–Muller codes, were developed in [1089, 1887]; see also Example 16.4.11 and Section 2.8. We define binary Reed–Muller codes recursively based on the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction; see [1323]. Other constructions of Reed–Muller codes can be found in Chapters 2, 16, and 20.

**Definition 1.11.1** For $i \in \{1, 2\}$, let $\mathcal{C}_i$ be linear codes both of length $n$ over $\mathbb{F}_q$. The $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ **construction** produces the linear code $\mathcal{C}$ of length $2n$ given by $\mathcal{C} = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \mid \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$.

**Remark 1.11.2** Let $\mathcal{C}_i$, for $i \in \{1, 2\}$, be $[n, k_i, d_i]_q$ codes with generator and parity check matrices $G_i$ and $H_i$, respectively. $\mathcal{C}$ obtained by the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction is a $[2n, k_1+k_2, \min\{2d_1, d_2\}]_q$ code with generator and parity check matrices

$$G = \left[\begin{array}{c|c} G_1 & G_1 \\ \hline \mathbf{0}_{k_2 \times n} & G_2 \end{array}\right] \quad \text{and} \quad H = \left[\begin{array}{c|c} H_1 & \mathbf{0}_{(n-k_1)\times n} \\ \hline -H_2 & H_2 \end{array}\right]. \tag{1.2}$$

We now define the binary Reed–Muller codes.

**Definition 1.11.3** Let $r$ and $m$ be integers with $0 \le r \le m$ and $1 \le m$. The $r^{\text{th}}$ **order binary Reed–Muller (RM) code of length** $2^m$, denoted $\mathcal{RM}(r, m)$, is defined recursively. The code $\mathcal{RM}(0, m) = \{\mathbf{0}, \mathbf{1}\}$, the $[2^m, 1, 2^m]_2$ **binary repetition code**, and $\mathcal{RM}(m, m) = \mathbb{F}_q^{2^m}$, a $[2^m, 2^m, 1]_2$ code. For $1 \le r < m$, define

$$\mathcal{RM}(r, m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \mid \mathbf{u} \in \mathcal{RM}(r, m-1), \mathbf{v} \in \mathcal{RM}(r-1, m-1)\}.$$

**Remark 1.11.4** Let $G(r,m)$ be a generator matrix of $\mathcal{RM}(r,m)$. By Definition 1.11.3, $G(0,m) = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$ and $G(m,m) = I_{2^m}$. By Definition 1.11.3 and (1.2), for $1 \leq r < m$,

$$G(r,m) = \left[\begin{array}{c|c} G(r,m-1) & G(r,m-1) \\ \hline O & G(r-1,m-1) \end{array}\right]$$

where $O = \mathbf{0}_{k \times 2^{m-1}}$ with $k$ the dimension of $\mathcal{RM}(r-1,m-1)$.

**Example 1.11.5** We give generator matrices for $\mathcal{RM}(r,m)$ with $1 \leq r < m \leq 3$:

$$G(1,2) = \left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 \end{array}\right], \quad G(1,3) = \left[\begin{array}{cccc|cccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right],$$

$$G(2,3) = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right].$$

From these generator matrices, we see that $\mathcal{RM}(1,2)$ and $\mathcal{RM}(2,3)$ consist of all even weight binary vectors of lengths 4 and 8, respectively. Also $\mathcal{RM}(1,3)$ is an $[8,4,4]_2$ code, which by Example 1.8.5 must be $\widehat{\mathcal{H}}_{3,2}$.

Using the definition of Reed–Muller codes and properties from the $(\mathbf{u} \mid \mathbf{u}+\mathbf{v})$ construction, along with induction, the following hold; see [1008, Theorem 1.10.1].

**Theorem 1.11.6** *Let $r$ and $m$ be integers with $0 \leq r \leq m$ and $1 \leq m$. The following hold.*

(a) $\mathcal{RM}(i,m) \subseteq \mathcal{RM}(j,m)$ *if* $0 \leq i \leq j \leq m$.

(b) *The dimension of* $\mathcal{RM}(r,m)$ *equals* $\binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r}$.

(c) *The minimum weight of* $\mathcal{RM}(r,m)$ *equals* $2^{m-r}$.

(d) $\mathcal{RM}(m,m)^{\perp} = \{\mathbf{0}\}$, *and if* $0 \leq r < m$, *then* $\mathcal{RM}(r,m)^{\perp} = \mathcal{RM}(m-r-1,m)$.

**Remark 1.11.7** Theorem 1.11.6(a) is sometimes called the **nesting property** of Reed–Muller codes. As observed in Example 1.11.5, $\mathcal{RM}(1,3) = \widehat{\mathcal{H}}_{3,2}$. Using Theorem 1.11.6(d), it can be shown that $\mathcal{RM}(m-2,m) = \widehat{\mathcal{H}}_{m,2}$; see [1008, Exercise 61]. By Theorem 1.11.6(d), $\mathcal{RM}(m-1,m) = \mathcal{RM}(0,m)^{\perp}$. Since $\mathcal{RM}(0,m) = \{\mathbf{0},\mathbf{1}\}$, $\mathcal{RM}(m-1,m)$ must be all even weight vectors in $\mathbb{F}_2^{2^m}$, a fact observed for $m=2$ and $m=3$ in Example 1.11.5.

## 1.12 Cyclic Codes

The study of cyclic codes seems to have begun with a series of four Air Force Cambridge Research Laboratory (AFCRL) technical notes [1536, 1537, 1538, 1539] by E. Prange from

1957 to 1959. The 1961 book by W. W. Peterson [1505] compiled extensive results about cyclic codes and laid the framework for much of the present-day theory. In 1972 this book was expanded and published jointly by Peterson and E. J. Weldon [1506].

Up to this point, the coordinates of $\mathbb{F}_q^n$ have been denoted $\{1, 2, \ldots, n\}$. For cyclic codes, the coordinates of $\mathbb{F}_q^n$ will be denoted $\{0, 1, \ldots, n-1\}$.

**Definition 1.12.1** Let $\mathcal{C}$ be a code of length $n$ over $\mathbb{F}_q$. $\mathcal{C}$ is **cyclic** provided that for all $\mathbf{c} = c_0 c_1 \cdots c_{n-1} \in \mathcal{C}$, the cyclic shift $\mathbf{c}' = c_{n-1} c_0 \cdots c_{n-2} \in \mathcal{C}$.

**Remark 1.12.2** The *cyclic shift* described in Definition 1.12.1 is cyclic shift to the *right* by one position with wrap-around. The code $\mathcal{C}$ is cyclic if and only if $P \in \mathrm{PAut}(\mathcal{C})$ where the permutation matrix $P = [p_{i,j}]$ is defined by $p_{i,i+1} = 1$ for $0 \leq i \leq n-2$, $p_{n-1,0} = 1$, and $p_{i,j} = 0$ otherwise. Cyclic codes are closed under cyclic shifts with wrap-around of *any* amount and in *either* the *left* or *right* directions.

**Example 1.12.3** Let $\mathcal{C}$ be the $[7, 4]_2$ code with generator matrix

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

Labeling the rows of $G$ as $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4$ top to bottom, we see that $\mathbf{r}_5 = 1000110 = \mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3$, $\mathbf{r}_6 = 0100011 = \mathbf{r}_2 + \mathbf{r}_3 + \mathbf{r}_4$, and $\mathbf{r}_7 = 1010001 = \mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_4$. Since $\mathcal{C}$ is spanned by $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_7\}$ and this list is closed under cyclic shifts, $\mathcal{C}$ must be a cyclic code. By row reducing $G$, we obtain another generator matrix

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

Label the columns of $G'$ left to right as $0, 1, \ldots, 6$. If $P$ is the $7 \times 7$ permutation matrix induced by the permutation that sends column 0 to column 2, column 2 to column 3, column 3 to column 1, column 1 to column 0, and fixes columns 4, 5, and 6, then $G'P$ has the same rows as $G_{3,2}$, the generator matrix of $\mathcal{H}_{3,2}$ in Example 1.4.9. Therefore, by ordering the coordinates of $\mathcal{H}_{3,2}$ appropriately, we see that $\mathcal{H}_{3,2}$ is a cyclic code.

While cyclic codes can be nonlinear, throughout this section we will examine only those that are linear. To study linear cyclic codes it is useful to consider elements of $\mathbb{F}_q^n$ as polynomials inside a certain quotient ring of polynomials. In that framework, linear cyclic codes are precisely the ideals of that quotient ring. We now establish the framework.

**Definition 1.12.4** Let $R$ be a commutative ring with identity. A subset $I$ of $R$ is an **ideal of** $R$ if for all $a, b \in I$ and $r \in R$, then $a - b \in I$ and $ra \in I$. The ideal $I$ is a **principal ideal** if there exists $a \in I$ such that $I = \{ra \mid r \in R\}$; $a$ is a **generator** of $I$ and $I$ is denoted $\langle a \rangle$.[6] The ring $R$ is an **integral domain** if whenever $a, b \in R$ and $ab = 0$, either $a = 0$ or $b = 0$. $R$ is a **principal ideal domain (PID)** if it is an integral domain and all its ideals are principal. The **quotient ring of** $R$ **by the ideal** $I$, denoted $R/I$, is the set of cosets $\{a + I \mid a \in R\}$ with addition and multiplication of cosets given by $(r + I) + (s + I) = (r + s) + I$ and $(r + I)(s + I) = rs + I$; two cosets $a + I$ and $b + I$ are equal if and only if $a - b \in I$.

[6] In some chapters of this books, the principal ideal generated by $a$ will be denoted $(a)$.

**Definition 1.12.5** Let $x$ be an indeterminate over $\mathbb{F}_q$. The set

$$\mathbb{F}_q[x] = \{a_0 + a_1x + \cdots + a_mx^m \mid a_i \in \mathbb{F}_q \text{ for } 0 \leq i \leq m \text{ and some } m\}$$

is the **ring of polynomials with coefficients in** $\mathbb{F}_q$. Let $a(x) = a_0 + a_1x + \cdots + a_mx^m \in \mathbb{F}_q[x]$ be a nonzero polynomial where $a_m \neq 0$. The **degree of** $a(x)$, denoted $\deg(a(x))$, is $m$. If $a_m = 1$, $a(x)$ is **monic**. If $\deg(a(x)) \geq 1$ and there do not exist $b(x), c(x) \in \mathbb{F}_q[x]$ with $\deg(b(x)) < \deg(a(x))$ and $\deg(c(x)) < \deg(a(x))$ such that $a(x) = b(x)c(x)$, then $a(x)$ is **irreducible over** $\mathbb{F}_q$.

The following result is standard; see for example [751].

**Theorem 1.12.6** *The following hold.*

(a) $\mathbb{F}_q[x]$ *is a commutative ring with identity* 1.

(b) (Division Algorithm) *For* $a(x), b(x) \in \mathbb{F}_q[x]$ *with* $b(x)$ *nonzero, there exists unique* $s(x), t(x) \in \mathbb{F}_q[x]$ *such that* $a(x) = b(x)s(x)+t(x)$ *where either* $t(x) = 0$ *or* $\deg(t(x)) < \deg(b(x))$.

(c) $\mathbb{F}_q[x]$ *is a PID.*

(d) (Unique Factorization) *Let* $p(x) \in \mathbb{F}_q[x]$ *with* $\deg(p(x)) \geq 1$. *There exists a unique set* $\{f_1(x), f_2(x), \ldots, f_t(x)\} \subseteq \mathbb{F}_q[x]$, *a unique list* $n_1, n_2, \ldots, n_t$ *of positive integers, and a unique* $\alpha \in \mathbb{F}_q$ *where each* $f_i(x)$ *is monic and irreducible over* $\mathbb{F}_q$ *such that*

$$p(x) = \alpha f_1(x)^{n_1} f_2(x)^{n_2} \cdots f_t(x)^{n_t}.$$

(e) (Unique Coset Representatives) *Let* $p(x) \in \mathbb{F}_q[x]$ *be nonzero. The distinct cosets of the quotient ring* $\mathbb{F}_q[x]/\langle p(x)\rangle$ *are uniquely representable as* $a(x) + \langle p(x)\rangle$ *where* $a(x) = 0$ *or* $\deg(a(x)) < \deg(p(x))$; $\mathbb{F}_q[x]/\langle p(x)\rangle$ *has order* $q^{\deg(p(x))}$. *The quotient ring* $\mathbb{F}_q[x]/\langle p(x)\rangle$ *is also a vector space over* $\mathbb{F}_q$ *of dimension* $\deg(p(x))$.

(f) *If* $p(x)$ *is irreducible over* $\mathbb{F}_q$, *then* $\mathbb{F}_q[x]/\langle p(x)\rangle$ *is a field.*

The map $\mathbf{a} = a_0a_1 \cdots a_{n-1} \mapsto a(x) + \langle x^n - 1\rangle$ where $a(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ is a vector space isomorphism from $\mathbb{F}_q^n$ onto $\mathbb{F}_q[x]/\langle x^n - 1\rangle$. We denote this map by $\mathbf{a} \mapsto a(x)$, dropping the '$+ \langle x^n - 1\rangle$'. Thus a linear code $\mathcal{C}$ of length $n$ can be viewed equivalently as a subspace of $\mathbb{F}_q^n$ or as an $\mathbb{F}_q$-subspace of $\mathbb{F}_q[x]/\langle x^n - 1\rangle$. Notice that if $\mathbf{a} \mapsto a(x)$, then $\mathbf{a}' = a_{n-1}a_0 \cdots a_{n-2} \mapsto xa(x)$ as $x^n + \langle x^n - 1\rangle = 1 + \langle x^n - 1\rangle$. So $\mathcal{C}$ is a cyclic code in $\mathbb{F}_q^n$ if and only if $\mathcal{C} \mapsto I$ where $I$ is an ideal of $\mathbb{F}_q[x]/\langle x^n - 1\rangle$. Therefore we study cyclic codes as ideals of $\mathbb{F}_q[x]/\langle x^n - 1\rangle$.

To find the ideals of $\mathbb{F}_q[x]/\langle x^n - 1\rangle$ requires factorization of $x^n - 1$. From the theory of finite fields (see [170, 1254, 1362]), there is an extension field of $\mathbb{F}_q$ that contains all the roots of $x^n - 1$. The smallest such field, called a **splitting field of** $x^n - 1$ **over** $\mathbb{F}_q$, is $\mathbb{F}_{q^t}$ where $t$ is the smallest integer such that $n \,|\, (q^t - 1)$. When $\gcd(n, q) = 1$, there exists $\alpha \in \mathbb{F}_{q^t}$, called a **primitive** $n^{\text{th}}$ **root of unity**, such that the $n$ distinct roots of $x^n - 1$ (called the **roots of unity**) are $\alpha^0 = 1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$; alternately if $\gamma$ is a primitive element of $\mathbb{F}_{q^t}$, one choice for $\alpha$ is $\gamma^{(q^t-1)/n}$. When $\gcd(n, q) \neq 1$, $x^n - 1$ has repeated roots. **For the remainder of this section, we assume** $\gcd(n, q) = 1$.[7]

[7]The theory of cyclic codes when $\gcd(n, q) \neq 1$ has some overlap with the theory when $\gcd(n, q) = 1$, but there are significant differences. When $\gcd(n, q) \neq 1$, cyclic codes are called **repeated-root cyclic codes**. Repeated-root cyclic codes were first examined in their most generality in [369, 1835].

**Definition 1.12.7** Let $s$ be an integer with $0 \leq s < n$. The **$q$-cyclotomic coset of $s$ modulo $n$** is the set

$$C_s = \{s, sq, \ldots, sq^{r-1}\} \bmod n$$

where $r$ is the smallest positive integer such that $sq^r \equiv s \pmod{n}$. The distinct $q$-cyclotomic cosets modulo $n$ partition the set of integers $\{0, 1, 2, \ldots, n-1\}$.

**Remark 1.12.8** A splitting field of $x^n - 1$ over $\mathbb{F}_q$ is $\mathbb{F}_{q^t}$ where $t$ is the size of the $q$-cyclotomic coset of 1 modulo $n$.

**Theorem 1.12.9 ([1323, Chapter 7.5])** *Let $\alpha$ be a primitive $n^{th}$ root of unity in the splitting field $\mathbb{F}_{q^t}$ of $x^n - 1$ over $\mathbb{F}_q$. For $0 \leq s < n$ define $M_{\alpha^s}(x) = \prod_{i \in C_s}(x - \alpha^i)$. Then $M_{\alpha^s}(x) \in \mathbb{F}_q[x]$ and is irreducible over $\mathbb{F}_q$. Furthermore, the unique factorization of $x^n - 1$ into monic irreducible polynomials over $\mathbb{F}_q$ is given by $x^n - 1 = \prod_s M_{\alpha^s}(x)$ where $s$ runs through a set of representatives of all distinct $q$-cyclotomic cosets modulo $n$.*

**Example 1.12.10** The 2-cyclotomic cosets modulo 7 are $C_0 = \{0\}$, $C_1 = \{1, 2, 4\}$, and $C_3 = \{3, 6, 5\}$. By Remark 1.12.8, $\mathbb{F}_{2^3} = \mathbb{F}_8$ is the splitting field of $x^7 - 1$ over $\mathbb{F}_2$. In the notation of Table 1.3, $\alpha = \delta$ is a primitive $7^{\text{th}}$ root of unity. In the notation of Theorem 1.12.9, $M_{\alpha^0}(x) = -1 + x = 1 + x$, $M_\alpha = (x - \alpha)(x - \alpha^2)(x - \alpha^4) = 1 + x + x^3$, $M_{\alpha^3} = (x - \alpha^3)(x - \alpha^6)(x - \alpha^5) = 1 + x^2 + x^3$, and $x^7 - 1 = M_{\alpha^0}(x)M_\alpha(x)M_{\alpha^3}(x)$.

Using Theorem 1.12.9, we have the following basic theorem [1008, Theorem 4.2.1] describing the structure of cyclic codes over $\mathbb{F}_q$. We remark that all of this theorem except part (g) is valid when $\gcd(n, q) \neq 1$. We note that if $a(x), b(x) \in \mathbb{F}_q[x]$, then $a(x)$ **divides** $b(x)$, denoted $a(x) \,|\, b(x)$, means that there exists $c(x) \in \mathbb{F}_q[x]$ such that $b(x) = a(x)c(x)$.

**Theorem 1.12.11** *Let $\mathcal{C}$ be a nonzero linear cyclic code over $\mathbb{F}_q$ of length $n$ viewed as an ideal of $\mathbb{F}_q[x]/\langle x^n - 1\rangle$. There exists a polynomial $g(x) \in \mathcal{C}$ with the following properties.*

(a) *$g(x)$ is the unique monic polynomial of minimum degree in $\mathcal{C}$.*

(b) *$\mathcal{C} = \langle g(x)\rangle$ in $\mathbb{F}_q[x]/\langle x^n - 1\rangle$.*

(c) *$g(x) \,|\, (x^n - 1)$.*

*With $k = n - \deg(g(x))$, let $g(x) = \sum_{i=0}^{n-k} g_i x^i$ where $g_{n-k} = 1$. Then*

(d) *the dimension of $\mathcal{C}$ is $k$ and $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ is a basis for $\mathcal{C}$,*

(e) *every element of $\mathcal{C}$ is uniquely expressible as a product $g(x)f(x)$ where $f(x) = 0$ or $\deg(f(x)) < k$,*

(f) *a generator matrix $G$ of $\mathcal{C}$ is*

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & \cdots & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & 0 & g_0 & \cdots & \cdots & \cdots & g_{n-k} \end{bmatrix}$$

$$\leftrightarrow \begin{bmatrix} g(x) & & & & \\ & xg(x) & & & \\ & & \ddots & & \\ & & & x^{k-1}g(x) \end{bmatrix},$$

*and*

(g) *if $\alpha$ is a primitive $n^{th}$ root of unity in the splitting field $\mathbb{F}_{q^t}$ of $x^n - 1$ over $\mathbb{F}_q$, then*

$$g(x) = \prod_s M_{\alpha^s}(x)$$

*where the product is over a subset of representatives of distinct $q$-cyclotomic cosets modulo $n$.*

**Definition 1.12.12** The polynomial $g(x)$ in Theorem 1.12.11 is the **generator polynomial** of $\mathcal{C}$. By convention, the cyclic code $\mathcal{C} = \{\mathbf{0}\}$ has generator polynomial $g(x) = x^n - 1$.

The following are immediate consequences of Theorem 1.12.11.

**Corollary 1.12.13** *There are $2^m$ linear cyclic codes of length $n$ (including the zero code) over $\mathbb{F}_q$ where $m$ is the number of $q$-cyclotomic cosets modulo $n$.*

**Corollary 1.12.14** *If $g_1(x)$ and $g_2(x)$ are generator polynomials of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, and if $g_1(x) \mid g_2(x)$, then $\mathcal{C}_2 \subseteq \mathcal{C}_1$.*

By Theorem 1.12.11, when $\gcd(n, q) = 1$, a linear cyclic code of length $n$ over $\mathbb{F}_q$ is uniquely determined by its generator polynomial. This in turn is determined by its roots in the splitting field of $x^n - 1$ over $\mathbb{F}_q$. This leads to the following definition.

**Definition 1.12.15** Let $\mathcal{C}$ be a linear cyclic code of length $n$ over $\mathbb{F}_q$ with generator polynomial $g(x) \mid (x^n - 1)$. Let $\alpha$ be a fixed primitive $n^{\text{th}}$ root of unity in a splitting field of $x^n - 1$ over $\mathbb{F}_q$. By Theorems 1.12.9 and 1.12.11, $g(x) = \prod_s \prod_{i \in C_s} (x - \alpha^i)$ where $s$ runs through some subset of representatives of the $q$-cyclotomic cosets $C_s$ modulo $n$. Let $T = \bigcup_s C_s$ be the union of these $q$-cyclotomic cosets. The roots of unity $\{\alpha^i \mid i \in T\}$ are called the **zeros** of $\mathcal{C}$; $\{\alpha^i \mid 0 \leq i < n,\ i \notin T\}$ are the **nonzeros** of $\mathcal{C}$. The set $T$ is called the **defining set of $\mathcal{C}$ relative to $\alpha$**.

**Remark 1.12.16** In Definition 1.12.15, if you change the primitive $n^{\text{th}}$ root of unity, you change the defining set $T$; so $T$ is computed relative to a fixed primitive root of unity.

**Remark 1.12.17** Corollary 1.12.14 can be translated into the language of defining sets: If $T_1$ and $T_2$ are defining sets of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, relative to the same primitive root of unity, and if $T_1 \subseteq T_2$, then $\mathcal{C}_2 \subseteq \mathcal{C}_1$.

**Example 1.12.18** Continuing with Example 1.12.10, Table 1.5 describes the $2^3 = 8$ binary cyclic codes of length 7. The code with $g(x) = 1 + x + x^3$ is $\mathcal{H}_{3,2}$ as discussed in Example 1.12.3. The code with $g(x) = 1 + x^2 + x^3$ is permutation equivalent to $\mathcal{H}_{3,2}$. The code of dimension $k = 1$ is the binary repetition code $\{\mathbf{0},\mathbf{1}\}$.

The dual code of a cyclic code is also cyclic. We can determine its generator polynomial and defining set; see [1323, Chapter 7.4].

**Theorem 1.12.19** *Let $\mathcal{C}$ be an $[n, k]_q$ cyclic code with generator polynomial $g(x)$. Define $h(x) = \dfrac{x^n - 1}{g(x)}$. Then $\mathcal{C}^\perp$ is cyclic with generator polynomial $g^\perp(x) = \dfrac{x^k h(x^{-1})}{h(0)}$. Let $\alpha$ be a primitive $n^{th}$ root of unity in a splitting field of $x^n - 1$ over $\mathbb{F}_q$. If $T$ is the defining set of $\mathcal{C}$ relative to $\alpha$, the defining set of $\mathcal{C}^\perp$ is $T^\perp = \{0, 1, \ldots, n-1\} \setminus (-1)T \bmod n$.*

**TABLE 1.5**: The $[7, k, d]_2$ cyclic codes with generator polynomial $g(x)$ and defining set $T$ relative to $\alpha$

| $k$ | $d$ | $g(x)$ | $T$ |
|---|---|---|---|
| 0 | — | $1 + x^7 = 0$ | $\{0, 1, 2, 3, 4, 5, 6\}$ |
| 1 | 7 | $1 + x + x^2 + x^3 + x^4 + x^5 + x^6$ | $\{1, 2, 3, 4, 5, 6\}$ |
| 3 | 4 | $1 + x^2 + x^3 + x^4$ | $\{0, 1, 2, 4\}$ |
| 3 | 4 | $1 + x + x^2 + x^4$ | $\{0, 3, 5, 6\}$ |
| 4 | 3 | $1 + x + x^3$ | $\{1, 2, 4\}$ |
| 4 | 3 | $1 + x^2 + x^3$ | $\{3, 5, 6\}$ |
| 6 | 2 | $1 + x$ | $\{0\}$ |
| 7 | 1 | 1 | $\emptyset$ |

**Example 1.12.20** Continuing with Example 1.12.18, the dual of any code in Table 1.5 must be a code in the table. By comparing dimensions, the codes of dimension 0 and 7 in the table are duals of each other as are the codes of dimension 1 and 6; this is confirmed by examining the defining sets and using Theorem 1.12.19. By this theorem, $\{0, 1, 2, 3, 4, 5, 6\} \setminus (-1)\{1, 2, 4\} \bmod 7 = \{0, 1, 2, 3, 4, 5, 6\} \setminus \{6, 5, 3\} = \{0, 1, 2, 4\}$ showing that the codes with defining sets $\{0, 1, 2, 4\}$ and $\{1, 2, 4\}$ are duals of each other. By Remark 1.12.17, as $\{1, 2, 4\} \subseteq \{0, 1, 2, 4\}$, $\langle 1+x+x^3\rangle^\perp = \langle 1+x^2+x^3+x^4\rangle \subseteq \langle 1+x+x^3\rangle$, a fact we already observed in Example 1.6.10. Similarly, the codes with defining sets $\{0, 3, 5, 6\}$ and $\{3, 5, 6\}$ are duals of each other.

The following is a somewhat surprising fact about cyclic self-orthogonal binary codes; see [1008, Theorem 4.4.18].

**Theorem 1.12.21** *A cyclic self-orthogonal binary code is doubly-even.*

**Example 1.12.22** As detailed in Examples 1.6.10 and 1.12.20, $\mathcal{H}_{3,2}^\perp$ is self-orthogonal and doubly-even, illustrating Theorem 1.12.21.

**Definition 1.12.23** Quasi-cyclic codes are a natural generalization of cyclic codes. Let $\mathcal{C}$ be a code of length $n$ and $\ell$ a positive integer dividing $n$. $\mathcal{C}$ is **$\ell$-quasi-cyclic** provided whenever $c_0c_1 \cdots c_{n-1} \in \mathcal{C}$ then $c_{n-\ell}c_{n-\ell+1} \cdots c_{n-1}c_0 \cdots c_{n-\ell-2}c_{n-\ell-1} \in \mathcal{C}$. Cyclic codes are 1-quasi-cyclic codes. Quasi-cyclic codes will be studied in Chapter 7.

See Chapter 2 and Sections 8.6 and 20.5 for more on cyclic codes over fields.

## 1.13 Golay Codes

In the same remarkable one-half page 1949 paper [820] in which Golay generalized the Hamming codes, he also introduced what later became known as the $[23, 12, 7]_2$ binary Golay code and the $[11, 6, 5]_3$ ternary Golay code. There are many ways to present these two codes; one way is to describe them as cyclic codes.

**Example 1.13.1** There are three 2-cyclotomic cosets modulo 23 with sizes 1, 11, and 11: $C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$, $C_5 = \{5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14\}$. Theorem 1.12.9 implies that, over $\mathbb{F}_2$, $x^{23} - 1 = x^{23} + 1$ factors into 3 monic irreducible

polynomials of degrees 1, 11, and 11. These irreducible factors are $b_0(x) = 1 + x$, $b_1(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11}$, and $b_2(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$. There are 8 binary linear cyclic codes of length 23 by Corollary 1.12.13. By Theorem 1.12.11(d), the codes $\mathcal{C}_1 = \langle b_1(x) \rangle$ and $\mathcal{C}_2 = \langle b_2(x) \rangle$ are $[23, 12]_2$ codes. The map that fixes coordinate 0 and switches coordinates $i$ and $23 - i$ for $1 \leq i \leq 11$ leads to a permutation matrix $P$ where $\mathcal{C}_1 P = \mathcal{C}_2$. Any code permutation equivalent to $\mathcal{C}_1$ is termed the $[23, 12]_2$ **binary Golay code of length 23** and is denoted $\mathcal{G}_{23}$. The splitting field of $x^{23} - 1$ over $\mathbb{F}_2$ is $\mathbb{F}_{2^{11}}$. In $\mathbb{F}_{2^{11}}$ there is a primitive $23^{\text{rd}}$ root of unity $\alpha$ where the defining sets of $\mathcal{C}_1$ and $\mathcal{C}_2$ are $C_1$ and $C_5$, respectively, relative to $\alpha$. Another primitive $23^{\text{rd}}$ root of unity is $\beta = \alpha^5$; relative to $\beta$, the defining sets of $\mathcal{C}_1$ and $\mathcal{C}_2$ are $C_5$ and $C_1$, respectively. By Theorem 1.12.19 the defining set of $\mathcal{C}_1^{\perp}$ relative to $\alpha$ is $C_0 \cup C_1$ implying by Remark 1.12.17 that $\mathcal{C}_1^{\perp} \subseteq \mathcal{C}_1$ showing $\mathcal{C}_1^{\perp}$ is self-orthogonal. Using Theorem 1.12.21, $\mathcal{C}_1^{\perp}$ is the doubly-even $[23, 11]_2$ code consisting of all codewords in $\mathcal{C}_1$ of even weight.

**Example 1.13.2** The 3-cyclotomic cosets modulo 11 are $C_0 = \{0\}$, $C_1 = \{1, 3, 9, 5, 4\}$, and $C_2 = \{2, 6, 7, 10, 8\}$ of sizes 1, 5, and 5, respectively. Theorem 1.12.9 implies that, over $\mathbb{F}_3$, $x^{11} - 1$ factors into 3 monic irreducible polynomials of degrees 1, 5, and 5. These irreducible factors are $t_0(x) = -1 + x$, $t_1(x) = -1 + x^2 - x^3 + x^4 + x^5$, and $t_2(x) = -1 - x + x^2 - x^3 + x^5$. There are 8 ternary linear cyclic codes of length 11 by Corollary 1.12.13. By Theorem 1.12.11(d), the codes $\mathcal{C}_1 = \langle t_1(x) \rangle$ and $\mathcal{C}_2 = \langle t_2(x) \rangle$ are $[11, 6]_3$ codes. The map that fixes coordinate 0 and switches coordinates $i$ and $11 - i$ for $1 \leq i \leq 5$ leads to a permutation matrix $P$ where $\mathcal{C}_1 P = \mathcal{C}_2$. Any code monomially equivalent to $\mathcal{C}_1$ is termed the $[11, 6]_3$ **ternary Golay code of length 11** and is denoted $\mathcal{G}_{11}$. The splitting field of $x^{11} - 1$ over $\mathbb{F}_3$ is $\mathbb{F}_{3^5}$. In $\mathbb{F}_{3^5}$ there is a primitive $11^{\text{th}}$ root of unity $\alpha$ where the defining sets of $\mathcal{C}_1$ and $\mathcal{C}_2$ are $C_1$ and $C_2$, respectively, relative to $\alpha$. Another primitive $11^{\text{th}}$ root of unity is $\beta = \alpha^2$; relative to $\beta$, the defining sets of $\mathcal{C}_1$ and $\mathcal{C}_2$ are $C_2$ and $C_1$, respectively.

**Definition 1.13.3** $\mathcal{G}_{23}$ can be extended as in Section 1.7 to a $[24, 12]_2$ code $\widehat{\mathcal{G}}_{23}$, denoted $\mathcal{G}_{24}$, and called the **binary Golay code of length 24**. Similarly $\mathcal{G}_{11}$ can be extended to a $[12, 6]_3$ code $\widehat{\mathcal{G}}_{11}$, denoted $\mathcal{G}_{12}$, and called the **ternary Golay code of length 12**.

**Remark 1.13.4** The automorphism groups of the four Golay codes involve the **Mathieu groups** $\mathrm{M}_p$ for $p \in \{11, 12, 23, 24\}$ discovered by Émile Mathieu [1352, 1353]. These four permutation groups on $p$ points are 4-fold transitive, when $p \in \{11, 23\}$, and 5-fold transitive, when $p \in \{12, 24\}$, simple groups. Properties of these groups and their relationship to Golay codes can be found in [442].

The following two theorems give basic properties of the four Golay codes. Parts (a), (b), and (c) of each theorem can be found in most standard coding theory texts. The uniqueness of these codes in part (d) of each theorem is a culmination of work in [525, 1514, 1518, 1732] with a self-contained proof in [1008, Chapter 10]. Part (e) of each theorem follows by direct computation from Definition 1.9.8. The automorphism groups in part (f) of each theorem can be found in [434], which is also [442, Chapter 10].

**Theorem 1.13.5** *The following properties hold for the binary Golay codes.*

(a) $\mathcal{G}_{23}$ *has minimum distance* 7 *and weight distribution* $A_0 = A_{23} = 1$, $A_7 = A_{16} = 253$, $A_8 = A_{15} = 506$, $A_{11} = A_{12} = 1288$, *and* $A_i = 0$ *otherwise.*

(b) $\mathcal{G}_{24}$ *has minimum distance* 8 *and weight distribution* $A_0 = A_{24} = 1$, $A_8 = A_{16} = 759$, $A_{12} = 2576$, *and* $A_i = 0$ *otherwise.*

(c) $\mathcal{G}_{24}$ *is doubly-even and self-dual.*

(d) *Both a* $(23, M)_2$ *and a* $(24, M)_2$*, possibly nonlinear, binary code each containing* $\mathbf{0}$ *with* $M \geq 2^{12}$ *codewords and minimum distance* 7 *and* 8*, respectively, are unique up to permutation equivalence. They are the* $[23, 12, 7]_2$ *and* $[24, 12, 8]_2$ *binary Golay codes.*

(e) $\mathcal{G}_{23}$ *is perfect.*

(f) $\mathrm{PAut}(\mathcal{G}_{23}) = \mathrm{M}_{23}$ *and* $\mathrm{PAut}(\mathcal{G}_{24}) = \mathrm{M}_{24}$.

**Theorem 1.13.6** *The following properties hold for the ternary Golay codes.*

(a) $\mathcal{G}_{11}$ *has minimum distance* 5 *and weight distribution* $A_0 = 1$, $A_5 = A_6 = 132$, $A_8 = 330$, $A_9 = 110$, $A_{11} = 24$, *and* $A_i = 0$ *otherwise.*

(b) $\mathcal{G}_{12}$ *has minimum distance* 6 *and weight distribution* $A_0 = 1$, $A_6 = 264$, $A_9 = 440$, $A_{12} = 24$, *and* $A_i = 0$ *otherwise.*

(c) $\mathcal{G}_{12}$ *is self-dual.*

(d) *Both a* $(11, M)_3$ *and a* $(12, M)_3$*, possibly nonlinear, ternary code each containing* $\mathbf{0}$ *with* $M \geq 3^6$ *codewords and minimum distance* 5 *and* 6*, respectively, are unique up to monomial equivalence. They are the* $[11, 6, 5]_3$ *and* $[12, 6, 6]_3$ *ternary Golay codes.*

(e) $\mathcal{G}_{11}$ *is perfect.*

(f) $\mathrm{MAut}(\mathcal{G}_{11}) = \widetilde{M}_{11}$ *and* $\mathrm{MAut}(\mathcal{G}_{12}) = \widetilde{M}_{12}$ *where* $\widetilde{M}_{11}$ *and* $\widetilde{M}_{12}$ *are isomorphic to the double covers, or the non-splitting central extensions by a center of order* 2*, of* $\mathrm{M}_{11}$ *and* $\mathrm{M}_{12}$.

**Remark 1.13.7** If there is equality for given parameters of a code in a bound from Section 1.9, we say the code **meets** the bound. Perfect codes are those meeting the Sphere Packing Bound; MDS codes are those meeting the Singleton Bound. Using Theorem 1.10.5, direct computation shows that the $[(q^m - 1)/(q - 1), m, q^{m-1}]_q$ simplex code meets the Griesmer Bound, as do $\mathcal{G}_{11}$ and $\mathcal{G}_{12}$. Neither $\mathcal{G}_{23}$ nor $\mathcal{G}_{24}$ meet the Griesmer Bound.

---

## 1.14 BCH and Reed–Solomon Codes

There is a lower bound, presented in Theorem 1.14.3, on the minimum distance of a cyclic code based on the defining set of the code. BCH codes take advantage of this bound. The binary BCH codes were discovered by A. Hocquenghem [968] and independently by R. C. Bose and D. K. Ray-Chaudhuri [248, 249], and were generalized to all finite fields by D. C. Gorenstein and N. Zierler [840]. Some properties of BCH codes are given in Section 2.6.

**Definition 1.14.1** Let $\mathcal{N} = \{0, 1, \ldots, n-1\}$ and $T \subseteq \mathcal{N}$. We say $T$ contains a set of $s \leq n$ **consecutive** elements provided there exists $b \in \mathcal{N}$ such that

$$\{b, b+1, \ldots, b+s-1\} \bmod n \subseteq T.$$

**Remark 1.14.2** When considering the notion of *consecutive*, wrap-around is allowed. For example if $n = 10$, $\{8, 9, 0, 1\}$ is a consecutive set in $T = \{0, 1, 2, 5, 8, 9\}$.

**TABLE 1.6**: The $[7, k, d]_2$ BCH codes with defining set $T$ relative to $\alpha$, $b$, and Bose distance $\delta$

| $k$ | $d$ | $T$ | $b$ | $\delta$ |
|---|---|---|---|---|
| 0 | — | $\{0,1,2,3,4,5,6\} = C_1 \cup C_2 \cup \cdots \cup C_6 \cup C_0$ | 1 | — |
| 1 | 7 | $\{1,2,3,4,5,6\} = C_1 \cup C_2 \cup \cdots \cup C_6$ | 1 | 7 |
| 3 | 4 | $\{0,1,2,4\} = C_0 \cup C_1 \cup C_2$ | 0 | 4 |
| 3 | 4 | $\{0,3,5,6\} = C_5 \cup C_6 \cup C_0$ | 5 | 4 |
| 4 | 3 | $\{1,2,4\} = C_1 \cup C_2$ | 1 | 3 |
| 4 | 3 | $\{3,5,6\} = C_5 \cup C_6$ | 5 | 3 |
| 6 | 2 | $\{0\} = C_0$ | 0 | 2 |

Rather surprisingly, the existence of consecutive elements in the defining set of a cyclic code determines a lower bound, called the *BCH Bound*, on the minimum distance of the code. A proof of the following can be found in [1323, Chapter 7.6].

**Theorem 1.14.3 (BCH Bound)** *Let $\mathcal{C}$ be a linear cyclic code of length $n$ over $\mathbb{F}_q$ and minimum distance $d$ with defining set $T$ relative to some primitive $n^{th}$ root of unity. Assume $T$ contains $\delta - 1$ consecutive elements for some integer $\delta \geq 2$. Then $d \geq \delta$.*

**Definition 1.14.4** Let $\delta$ be an integer with $2 \leq \delta \leq n$. A **BCH code over $\mathbb{F}_q$ of length $n$ and designed distance** $\delta$ is a linear cyclic code with defining set

$$T = C_b \cup C_{b+1} \cup \cdots \cup C_{b+\delta-2}$$

relative to some primitive $n^{\text{th}}$ root of unity where $C_i$ is the $q$-cyclotomic coset modulo $n$ containing $i$. As $T$ contains the consecutive set $\{b, b+1, \ldots, b+\delta-2\}$, this code has minimum distance at least $\delta$ by the BCH Bound. If $b = 1$, the code is **narrow-sense**; if $n = q^t - 1$ for some $t$, the code is **primitive**.

**Definition 1.14.5** Sometimes a BCH code can have more than one designed distance; the largest designed distance is called the **Bose distance**.

**Example 1.14.6** Consider the eight $[7, k, d]_2$ binary cyclic codes from Example 1.12.18 and presented in Table 1.5. All except the code with defining set $T = \emptyset$ are BCH codes as seen in Table 1.6. As $7 = 2^3 - 1$, all these BCH codes are primitive. Technically, the zero code is primitive with designed distance 8; of course distance in the zero code is meaningless. Of the six remaining codes, three are narrow-sense. Notice that the code with defining set $\{1, 2, 4\}$ is narrow-sense with two designed distances 2 and 3 as $\{1, 2, 4\} = C_1 = C_1 \cup C_2$; the Bose distance is 3. The code with defining set $\{1, 2, 3, 4, 5, 6\}$ is narrow-sense with designed distances 4 through 7 as $\{1, 2, 3, 4, 5, 6\} = C_1 \cup C_2 \cup C_3 = C_1 \cup C_2 \cup C_3 \cup C_4 = C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5 = C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5 \cup C_6$; the Bose distance is 7. The Bose designed distance and the true minimum distance are the same for the seven nonzero BCH codes.

**Example 1.14.7** In the notation of Example 1.13.1, $\mathcal{G}_{23}$ has defining set $T = C_1$ which contains 4 consecutive elements $\{1, 2, 3, 4\}$. By the BCH Bound, $\mathcal{G}_{23}$ has minimum weight at least 5; its true minimum weight is 7 from Theorem 1.13.5(a). As the defining set of $\mathcal{G}_{23}$ is $T = C_1 = C_1 \cup C_2 \cup C_3 \cup C_4$, $\mathcal{G}_{23}$ is a narrow-sense[8] BCH code of Bose designed distance

[8] $\mathcal{G}_{23}$ is permutation equivalent to the BCH code with designed distance 5 and defining set $C_5 = C_{19} = C_{19} \cup C_{20} \cup C_{21} \cup C_{22}$; in this formulation $\mathcal{G}_{23}$ is not narrow-sense.

$\delta = 5$ with $b = 1$. Similarly, $\mathcal{G}_{11}$ of Example 1.13.2 is a BCH code viewed in several ways. $\mathcal{G}_{11}$ is a narrow-sense BCH code with $b = 1$, $\delta = 2$ and defining set $C_1 = \{1, 3, 4, 5, 9\}$. It is also a BCH code with $b = 3$, $\delta = 2, 3$, or 4 as $\{1, 3, 4, 5, 9\} = C_3 = C_3 \cup C_4 = C_3 \cup C_4 \cup C_5$. The Bose distance of $\mathcal{G}_{11}$ is 4 while its true minimum distance is 5 from Theorem 1.13.6(a).

At about the same time as BCH codes appeared in the literature, I. S. Reed and G. Solomon [1582] published their work on the codes that now bear their names. These codes, which are now commonly presented as a special case of BCH codes, were actually first constructed by K. A. Bush [319] in 1952 in the context of orthogonal arrays. Because of their burst error-correction capabilities, Reed–Solomon codes are used to improve the reliability of compact discs, digital audio tapes, and other data storage systems.

**Definition 1.14.8** A **Reed–Solomon (RS) code**[9] **of length** $n$ **over** $\mathbb{F}_q$ is a primitive BCH code of length $n = q - 1$.

When $n = q - 1$, the $q$-cyclotomic coset modulo $n$ containing $s$ is $C_s = \{s\}$. So if $\mathcal{C}$ is an $[n, k, d]_q$ Reed–Solomon code, its defining set $T$ has size $n - k$ and must be $\{b, b+1, \ldots, b+(n-k-1)\}$ mod $n$ for some $b$. By the BCH Bound, $d \geq n - k + 1$. By the Singleton Bound, $d \leq n - k + 1$. Therefore $d = n - k + 1$ and $\mathcal{C}$ is MDS; in particular the designed distance $\delta = n - k + 1$ equals the true minimum distance. In general the dual code of an MDS code is also MDS by Theorem 1.9.13. The dual code of a BCH code may not be BCH; however the dual of a Reed–Solomon code is Reed–Solomon as follows. By Theorem 1.12.19, $\mathcal{C}^\perp$ has defining set $T^\perp = \mathcal{N} \setminus (-1)T$ mod $n$ where $\mathcal{N} = \{0, 1, \ldots, n-1\}$. Since $(-1)T$ mod $n$ consists of $n-k$ consecutive elements of $\mathcal{N}$, $T^\perp$ is the remaining $k$ elements $\mathcal{N}$, which clearly must be consecutive (recalling that *wrap-around* is allowed in consecutive sets modulo $n$). This discussion yields the following result.

**Theorem 1.14.9** *Let $\mathcal{C}$ be a Reed–Solomon code over $\mathbb{F}_q$ of length $n = q - 1$ and designed distance $\delta$. The following hold.*

(a) *$\mathcal{C}$ has defining set $T = \{b, b+1, \ldots, b+\delta-2\}$ for some integer $b$.*

(b) *$\mathcal{C}$ has minimum distance $d = \delta$ and dimension $k = n - d + 1$.*

(c) *$\mathcal{C}$ is MDS.*

(d) *$\mathcal{C}^\perp$ is a Reed–Solomon code of designed distance $k + 1$.*

**Example 1.14.10** Using Table 1.1, $\gamma$ is both a primitive element of $\mathbb{F}_9$ and a primitive $8^{\text{th}}$ root of unity. Let $\mathcal{C}$ be the narrow-sense Reed–Solomon code over $\mathbb{F}_9$ of length 8 and designed distance $\delta = 4$. Then $\mathcal{C}$ has defining set $\{1, 2, 3\}$ relative to $\gamma$ and generator polynomial $g(x) = (x-\gamma)(x-\gamma^2)(x-\gamma^3) = \gamma^2 + \gamma x + \gamma^3 x^2 + x^3$. $\mathcal{C}$ is an $[8, 5, 4]_9$ code. By Theorem 1.12.19, $\mathcal{C}^\perp$ has defining set $T^\perp = \{0, 1, \ldots, 7\} \setminus (-1)\{1, 2, 3\}$ mod $8 = \{0, 1, \ldots, 7\} \setminus \{7, 6, 5\} = \{0, 1, 2, 3, 4\}$ and hence generator polynomial $g^\perp(x) = (x-1)(x-\gamma)(x-\gamma^2)(x-\gamma^3)(x-\gamma^4) = (x^2-1)g(x) = \gamma^6 + \gamma^5 x + \gamma^5 x^2 + \gamma^7 x^3 + \gamma^3 x^4 + x^5$. So $\mathcal{C}^\perp$ is an $[8, 3, 6]_9$ non-narrow-sense Reed–Solomon code with $b = 0$ and designed distance 6, consistent with Theorem 1.14.9(d). As $T \subseteq T^\perp$, $\mathcal{C}^\perp \subseteq \mathcal{C}$ by Remark 1.12.17.

The original formulation of Reed and Solomon for the narrow-sense Reed–Solomon codes is different from that of Definition 1.14.8. This alternative formulation of narrow-sense Reed–Solomon codes is of particular importance because it is the basis for the definitions

[9] While this is a common definition of Reed–Solomon codes, there are other codes of lengths different from $q - 1$ that are also called Reed–Solomon codes. See Remark 15.3.21.

of generalized Reed–Solomon codes, Goppa codes, and algebraic geometry codes; see Chapters 15 and 24.

For this formulation, let $\mathcal{P}_{k,q} = \{p(x) \in \mathbb{F}_q[x] \mid p(x) = 0 \text{ or } \deg(p(x)) < k\}$ when $k \geq 0$. See [1008, Theorem 5.2.3] for a proof of the following.

**Theorem 1.14.11** *Let $n = q-1$ and let $\alpha$ be a primitive $n^{th}$ root of unity in $\mathbb{F}_q$. For $0 < k \leq n = q-1$, let $\mathcal{RS}_k(\boldsymbol{\alpha}) = \left\{\left(p(\alpha^0), p(\alpha), \ldots, p(\alpha^{q-2})\right) \in \mathbb{F}_q^n \mid p(x) \in \mathcal{P}_{k,q}\right\}$. Then $\mathcal{RS}_k(\boldsymbol{\alpha})$ is the narrow-sense $[q-1, k, q-k]_q$ Reed–Solomon code.*

In general, extending an MDS code may not produce an MDS code; however extending a narrow-sense Reed–Solomon code does produce an MDS code. With the notation of Theorem 1.14.11, $\mathbb{F}_q = \{0, 1 = \alpha^0, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$ and, when $q \geq 3$, $\sum_{i=0}^{q-2} \alpha^i = 0$. Using this, it is straightforward to show that if $q \geq 3$, $k < q-1$, and $p(x) \in \mathcal{P}_{k,q}$, then $\sum_{\beta \in \mathbb{F}_q} p(\beta) = 0$. This leads to the following result.

**Theorem 1.14.12** *With the notation of Theorem* 1.14.11 *and $0 < k < n = q-1$, $\widehat{\mathcal{RS}}_k(\boldsymbol{\alpha}) = \left\{\left(p(\alpha^0), p(\alpha), \ldots, p(\alpha^{q-2}), p(0)\right) \in \mathbb{F}_q^n \mid p(x) \in \mathcal{P}_{k,q}\right\}$ is a $[q, k, q-k+1]_q$ MDS code.*

**Remark 1.14.13** The code $\mathcal{RS}_{q-1}(\boldsymbol{\alpha})$ omitted from consideration in Theorem 1.14.12 equals $\mathbb{F}_q^{q-1}$. Its extension is not as given in Theorem 1.14.12; however $\widehat{\mathcal{RS}}_{q-1}(\boldsymbol{\alpha})$ is still a $[q, q-1, 2]_q$ MDS code.

---

## 1.15 Weight Distributions

The weight distribution of a linear code determines the weight distribution of its dual code via a series of equations, called the MacWilliams Identities or the MacWilliams Equations. They were first developed by F. J. MacWilliams in [1319]. There are in fact several equivalent formulations of these equations. Among these are the Pless Power Moments discovered by V. S. Pless [1513]. The most compact form of these identities is expressed in a single polynomial equation relating the weight distribution of a code and its dual.

**Definition 1.15.1** Let $\mathcal{C}$ be a linear code of length $n$ over $\mathbb{F}_q$ with weight distribution $A_i(\mathcal{C})$ for $0 \leq i \leq n$. Let $x$ and $y$ be independent indeterminates over $\mathbb{F}_q$. The **(Hamming) weight enumerator of** $\mathcal{C}$ is defined to be

$$\mathrm{Hwe}_{\mathcal{C}}(x, y) = \sum_{i=0}^{n} A_i(\mathcal{C}) x^i y^{n-i}.$$

The formulation of the Pless Power Moments involves Stirling numbers.

**Definition 1.15.2** The **Stirling numbers $S(r, \nu)$ of the second kind** are defined for nonnegative integers $r, \nu$ by the equation

$$S(r, \nu) = \frac{1}{\nu!} \sum_{i=0}^{\nu} (-1)^{\nu - i} \binom{\nu}{i} i^r;$$

$\nu! S(r, \nu)$ is the number of ways to distribute $r$ distinct objects into $\nu$ distinct boxes with no box left empty.

The next theorem gives six equivalent formulations of the **MacWilliams Identities** or **MacWilliams Equations**. The fourth in the list involves the Krawtchouck polynomials; see Definition 1.9.21. The last two are the **Pless Power Moments**. One proof of the equivalence of these identities is found in [1008, Chapter 7.2].

**Theorem 1.15.3 (MacWilliams Identities and Pless Power Moments)** *Let $\mathcal{C}$ be a linear $[n,k]_q$ code and $\mathcal{C}^\perp$ its $[n,n-k]_q$ dual code. Let $A_i = A_i(\mathcal{C})$ and $A_i^\perp = A_i(\mathcal{C}^\perp)$, for $0 \le i \le n$, be the weight distributions of $\mathcal{C}$ and $\mathcal{C}^\perp$, respectively. The following are equivalent.*

(a) $\displaystyle\sum_{j=0}^{n-\nu} \binom{n-j}{\nu} A_j = q^{k-\nu} \sum_{j=0}^{\nu} \binom{n-j}{n-\nu} A_j^\perp$ *for* $0 \le \nu \le n$.

(b) $\displaystyle\sum_{j=\nu}^{n} \binom{j}{\nu} A_j = q^{k-\nu} \sum_{j=0}^{\nu} (-1)^j \binom{n-j}{n-\nu} (q-1)^{\nu-j} A_j^\perp$ *for* $0 \le \nu \le n$.

(c) $\mathrm{Hwe}_{\mathcal{C}^\perp}(x,y) = \dfrac{1}{|\mathcal{C}|} \mathrm{Hwe}_{\mathcal{C}}(y-x, y+(q-1)x)$.

(d) $\displaystyle A_j^\perp = \frac{1}{|\mathcal{C}|} \sum_{i=0}^{n} A_i K_j^{(n,q)}(i)$ *for* $0 \le j \le n$.

(e) $\displaystyle\sum_{j=0}^{n} j^r A_j = \sum_{j=0}^{\min\{n,r\}} (-1)^j A_j^\perp \left( \sum_{\nu=j}^{r} \nu! S(r,\nu) q^{k-\nu} (q-1)^{\nu-j} \binom{n-j}{n-\nu} \right)$ *for* $0 \le r$.

(f) $\displaystyle\sum_{j=0}^{n} (n-j)^r A_j = \sum_{j=0}^{\min\{n,r\}} A_j^\perp \left( \sum_{\nu=j}^{r} \nu! S(r,\nu) q^{k-\nu} \binom{n-j}{n-\nu} \right)$ *for* $0 \le r$.

**Remark 1.15.4** In the case of a linear code, the Delsarte–MacWilliams Inequalities of Theorem 1.9.22 follow from Theorem 1.15.3(d).

**Example 1.15.5** In Theorem 1.13.6(b), we gave the weight distribution of $\mathcal{G}_{12}$. Using the Pless Power Moments of Theorem 1.15.3(e), we can verify this weight distribution using only the fact that $\mathcal{G}_{12}$ is a $[12,6,6]_3$ self-dual code. As $\mathcal{G}_{12}$ is self-dual, $A_i^\perp = A_i$ and $A_i = 0$ when $3 \nmid i$ by Theorem 1.6.2(h). As $A_0 = 1$ and the minimum weight of $\mathcal{G}_{12}$ is 6, the only unknown $A_i$ are $A_6$, $A_9$, and $A_{12}$. We can find these from the first three power moments in Theorem 1.15.3(e). For a general $[n,k]_q$ code, as $A_0^\perp = 1$, the first three power moments are

$$\sum_{j=0}^{n} A_j = q^k,$$
$$\sum_{j=0}^{n} j A_j = q^{k-1}(qn - n - A_1^\perp), \text{ and}$$
$$\sum_{j=0}^{n} j^2 A_j = q^{k-2}\big((q-1)n(qn-n+1) - (2qn - q - 2n + 2)A_1^\perp + 2A_2^\perp\big).$$

Applied specifically to $\mathcal{G}_{12}$, these become

$$1 + A_6 + A_9 + A_{12} = 729$$
$$6A_6 + 9A_9 + 12A_{12} = 5\,832$$
$$36A_6 + 81A_9 + 144A_{12} = 48\,600.$$

The unique solution to this system is $A_6 = 264$, $A_9 = 440$, and $A_{12} = 24$. Thus the weight enumerator of $\mathcal{G}_{12}$ is

$$\mathrm{Hwe}_{\mathcal{G}_{12}}(x, y) = y^{12} + 264x^6y^6 + 440x^9y^3 + 24x^{12}.$$

**Example 1.15.6** Let $\mathcal{C}$ be the $\left[(q^m-1)/(q-1), m, q^{m-1}\right]_q$ simplex code of Theorem 1.10.5; by this theorem, with $n = (q^m - 1)/(q - 1)$,

$$\mathrm{Hwe}_{\mathcal{C}}(x, y) = y^n + (q^m - 1)x^{q^{m-1}}y^{n-q^{m-1}}.$$

By Definition 1.10.1, $\mathcal{C}^\perp = \mathcal{H}_{m,q}$. Using Theorem 1.15.3(d),

$$A_j(\mathcal{H}_{m,q}) = \frac{1}{q^m}\left(K_j^{(n,q)}(0) + (q^m - 1)K_j^{(n,q)}(q^{m-1})\right) \quad \text{for } 0 \le j \le n$$

noting that

$$K_j^{(n,q)}(0) = (q-1)^j \binom{n}{j} \quad \text{and}$$

$$K_j^{(n,q)}(q^{m-1}) = \sum_{i=1}^{j} (-1)^i (q-1)^{j-i} \binom{q^{m-1}}{i} \binom{n - q^{m-1}}{j-i}.$$

The MacWilliams Identities can be used to find the weight distribution of an MDS code as found, for example, in [1323, Theorem 6 of Chapter 11]. A resulting corollary gives bounding relations on the length, dimension, and field size.

**Theorem 1.15.7** *Let $\mathcal{C}$ be an $[n, k, d]_q$ MDS code over $\mathbb{F}_q$. The weight distribution of $\mathcal{C}$ is given by $A_0(\mathcal{C}) = 1$, $A_i(\mathcal{C}) = 0$ for $1 \le i < d = n - k + 1$ and*

$$A_i(\mathcal{C}) = \binom{n}{i} \sum_{j=0}^{i-d} (-1)^j \binom{i}{j} \left(q^{i+1-d-j} - 1\right)$$

*for $d \le i \le n$.*

**Corollary 1.15.8** *Let $\mathcal{C}$ be an $[n, k, d]_q$ MDS code over $\mathbb{F}_q$.*

(a) *If $2 \le k$, then $d = n - k + 1 \le q$.*

(b) *If $k \le n - 2$, then $k + 1 \le q$.*

This corollary becomes a foundation for the MDS Conjecture 3.3.21 in Chapter 3.

---

## 1.16 Encoding

Figure 1.1 shows a simple communication channel that includes a component called an *encoder*, in which a message is encoded to produce a codeword. In this section we examine two encoding processes.

As in Figure 1.1, a message is any of the $q^k$ possible $k$-tuples $\mathbf{x} \in \mathbb{F}_q^k$. The encoder will convert $\mathbf{x}$ to an $n$-tuple $\mathbf{c}$ from a code $\mathcal{C}$ over $\mathbb{F}_q$ with $q^k$ codewords; that codeword will then be transmitted over the communication channel.

Suppose that $\mathcal{C}$ is an $[n,k,d]_q$ linear code with generator matrix $G$ and parity check matrix $H$. We first describe an encoder that uses the generator matrix $G$. The most common way to encode the message $\mathbf{x}$ is as $\mathbf{x} \mapsto \mathbf{c} = \mathbf{x}G$. If $G$ is replaced by another generator matrix, the encoding of $\mathbf{x}$ will, of course, be different. A nice relationship exists between message and codeword if $G$ is in standard form $\left[I_k \mid A\right]$. In that case the first $k$ coordinates of the codeword $\mathbf{c}$ are the information symbols $\mathbf{x}$ in order; the remaining $n-k$ symbols are the parity check symbols, that is, the redundancy added to $\mathbf{x}$ in order to help recover $\mathbf{x}$ if errors occur during transmission. A similar relationship between message and codeword can exist even if $G$ is not in standard form. Specifically, suppose there exist column indices $i_1, i_2, \ldots, i_k$ such that the $k \times k$ matrix consisting of these $k$ columns of $G$ is the $k \times k$ identity matrix. In that case the message is found in the $k$ coordinates $i_1, i_2, \ldots, i_k$ of the codeword scrambled but otherwise unchanged; that is, the message symbol $x_j$ is in component $i_j$ of the codeword. If this occurs where the message is embedded in the codeword, we say that the encoder is a **systematic encoder of** $\mathcal{C}$. We can always force an encoder to be systematic. For example, if $G$ is row reduced to a matrix $G_1$ in reduced row echelon form, $G_1$ remains a generator matrix of $\mathcal{C}$ by Remark 1.4.3; the encoding $\mathbf{x} \mapsto \mathbf{c} = \mathbf{x}G_1$ is systematic as $G_1$ has $k$ columns which together form $I_k$. Another way to force an encoder to be systematic is as follows. By Theorem 1.8.4, $\mathcal{C}$ is permutation equivalent to an $[n,k,d]_q$ code $\mathcal{C}'$ with generator matrix $G'$ in standard form. If the code $\mathcal{C}'$ is used in place of $\mathcal{C}$, the encoder $\mathbf{x} \mapsto \mathbf{x}G'$ is a systematic encoder of $\mathcal{C}'$.

**Example 1.16.1** Let $\mathcal{C}$ be the $[7,4,3]_2$ binary Hamming code $\mathcal{H}_{3,2}$ with generator matrix $G_{3,2}$ from Example 1.4.9. Encoding $\mathbf{x} = x_1x_2x_3x_4 \in \mathbb{F}_2^4$ as $\mathbf{x}G_{3,2}$ produces the codeword $\mathbf{c} = x_1x_2x_3x_4(x_2+x_3+x_4)(x_1+x_3+x_4)(x_1+x_2+x_4)$.

**Example 1.16.2** Let $\mathcal{C}$ be an $[n,k,d]_q$ cyclic code with generator polynomial $g(x)$ and generator matrix $G$ obtained from cyclic shifts of $g(x)$ as in Theorem 1.12.11(f). Suppose the message $\mathbf{m} = m_0m_1\cdots m_{k-1}$ is to be encoded as $\mathbf{c} = \mathbf{m}G$. Using the polynomial $m(x) = m_0+m_1x+\cdots+m_{k-1}x^{k-1}$ to represent the message $\mathbf{m}$ and $c(x) = c_0+c_1x+\cdots+c_{n-1}x^{n-1}$ to represent the codeword $\mathbf{c}$, it is a simple calculation to show $c(x) = m(x)g(x)$. Generally, this encoding is not systematic. Recall from Examples 1.12.3 and 1.12.18 that the Hamming $[7,4,3]_2$ code $\mathcal{H}_{3,2}$ has a cyclic form with generator polynomial $g(x) = 1+x+x^3$. The nonsystematic encoder $m(x) \mapsto c(x) = m(x)g(x)$ yields $c(x) = m_0 + (m_0+m_1)x + (m_1+m_2)x^2 + (m_0+m_2+m_3)x^3 + (m_1+m_3)x^4 + m_2x^5 + m_3x^6$.

The second method to encode uses the parity check matrix $H$. This is easiest to do when $G$ is in standard form $\left[I_k \mid A\right]$; in this case $H = \left[-A^{\mathsf{T}} \mid I_{n-k}\right]$ by Theorem 1.4.7. Suppose that $\mathbf{x} = x_1x_2\cdots x_k$ is to be encoded as the codeword $\mathbf{c} = c_1c_2\cdots c_n = \mathbf{x}G$. As $G$ is in standard form, $c_1c_2\cdots c_k = x_1x_2\cdots x_k$. So we need to determine the $n-k$ redundancy symbols $c_{k+1}c_{k+2}\cdots c_n$. Because $\mathbf{0}^{\mathsf{T}} = H\mathbf{c}^{\mathsf{T}} = \left[-A^{\mathsf{T}} \mid I_{n-k}\right]\mathbf{c}^{\mathsf{T}}$, we have $A^{\mathsf{T}}\mathbf{x}^{\mathsf{T}} = c_{k+1}c_{k+2}\cdots c_n^{\mathsf{T}}$, or equivalently $c_{k+1}c_{k+2}\cdots c_n = \mathbf{x}A$. This process can be generalized when $\mathbf{x} \mapsto \mathbf{x}G$ is a systematic encoder.

**Example 1.16.3** Continuing with Example 1.16.1, we can encode $\mathbf{x} = x_1x_2x_3x_4$ using $H_{3,2}$ from Example 1.4.9. Here $c_5c_6c_7 = \mathbf{x}A$ where

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

Thus $c_5 = x_2+x_3+x_4$, $c_6 = x_1+x_3+x_4$, and $c_7 = x_1+x_2+x_4$, consistent with Example 1.16.1.

**Example 1.16.4** Let $\mathcal{C}$ be an $[n,k,d]_q$ cyclic code with generator polynomial $g(x)$. In Example 1.16.2 a nonsystematic encoder was described that encodes a cyclic code using $g(x)$. There is a systematic encoder of $\mathcal{C}$ using the generator polynomial $g^{\perp}(x)$ of $\mathcal{C}^{\perp}$. By Theorem 1.12.19, $g^{\perp}(x) = x^k h(x^{-1})/h(0) = h'_0 + h'_1 x + \cdots + h'_{k-1}x^{k-1} + h'_k x^k$ where $h(x) = (x^n - 1)/g(x)$ and $h'_k = 1$. Let $H$, which is a parity check matrix for $\mathcal{C}$, be determined from the shifts of $g^{\perp}(x)$ as follows:

$$H = \begin{bmatrix} h'_0 & h'_1 & h'_2 & \cdots & h'_k & \cdots & \cdots & 0 \\ 0 & h'_0 & h'_1 & \cdots & h'_{k-1} & h'_k & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & 0 & h'_0 & \cdots & \cdots & \cdots & h'_k \end{bmatrix}$$

$$\leftrightarrow \begin{bmatrix} g^{\perp}(x) & & & & & & \\ & x g^{\perp}(x) & & & & & \\ & & & \ddots & & & \\ & & & & x^{n-k-1} g^{\perp}(x) & & \end{bmatrix}.$$

Examining the generator matrix $G$ for $\mathcal{C}$ in Theorem 1.12.11(f), $\{0, 1, \ldots, k-1\}$ is an information set for $\mathcal{C}$. Let $\mathbf{c} = c_0 c_1 \cdots c_{n-1} \in \mathcal{C}$; so $c_0 c_1 \cdots c_{k-1}$ can be considered the associated message. The redundancy components $c_k c_{k+1} \cdots c_{n-1}$ are determined from $H\mathbf{c}^{\mathsf{T}} = \mathbf{0}^{\mathsf{T}}$ and can be computed in the order $i = k, k+1, \ldots, n-1$ where

$$c_i = -\sum_{j=0}^{k-1} h'_j c_{i-k+j}. \tag{1.3}$$

**Example 1.16.5** We apply the systematic encoding of Example 1.16.4 to the cyclic version of the Hamming $[7,4,3]_2$ code $\mathcal{H}_{3,2}$ with generator polynomial $g(x) = 1 + x + x^3$; see Example 1.12.18. By Example 1.12.20, $g^{\perp}(x) = 1 + x^2 + x^3 + x^4$ and (1.3) yields $c_4 = c_0 + c_2 + c_3$, $c_5 = c_1 + c_3 + c_4$, and $c_6 = c_2 + c_4 + c_5$. In terms of the information bits $c_0 c_1 c_2 c_3$, we have $c_4 = c_0 + c_2 + c_3$, $c_5 = c_0 + c_1 + c_2$, and $c_6 = c_1 + c_2 + c_3$.

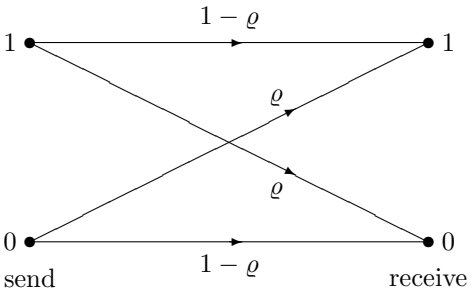
As discussed in Section 1.1, sometimes the receiver is interested only in the sent codewords rather than the sent messages. However, if there is interest in the actual message, a question arises as to how to recover the message from a codeword. If the encoder $\mathbf{x} \mapsto \mathbf{x}G$ is systematic, it is straightforward to recover the message. What can be done otherwise? Because $G$ has independent rows, there is an $n \times k$ matrix $K$ such that $GK = I_k$; $K$ is called a **right inverse for** $G$. A right inverse is not necessarily unique. As $\mathbf{c} = \mathbf{x}G$, the message $\mathbf{x} = \mathbf{x}GK = \mathbf{c}K$.

**Example 1.16.6** In Example 1.16.2, we encoded the message $m_0 m_1 m_2 m_3$ using the $[7,4,3]_2$ cyclic version of $\mathcal{H}_{3,2}$ with generator polynomial $g(x) = 1 + x + x^3$. The resulting codeword was $\mathbf{c} = (m_0, m_0 + m_1, m_1 + m_2, m_0 + m_2 + m_3, m_1 + m_3, m_2, m_5)$. The generator matrix $G$ obtained from $g(x)$ as in Theorem 1.12.11(f) has right inverse

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

Computing $\mathbf{c}K$ gives $m_0 m_1 m_2 m_3$ as expected.

**FIGURE 1.2**: Binary symmetric channel with crossover probability $\varrho$




## 1.17 Decoding

Decoding is the process of determining which codeword $\mathbf{c}$ was sent when a vector $\mathbf{y}$ is received. Decoding is generally more complex than encoding. Decoding algorithms usually vary with the type of code being used. In this section we discuss only the basics of hard-decision decoding. Decoding is discussed more in depth in Chapters 15, 21, 24, 28–30, and 32.

**Definition 1.17.1** A **hard-decision decoder** is a decoder that inputs 'hard' data from the channel (e.g., elements from $\mathbb{F}_q$) and outputs hard data to the receiver. A **soft-decision decoder** is one which inputs 'soft' data from the channel (e.g., estimates of the symbols with attached probabilities) and generally outputs hard data.

Initially we focus our discussion to the decoding of binary codes. To set the stage for decoding, we begin with one possible mathematical model of a channel that transmits binary data. Before stating the model, we establish some notation. If $E$ is an event, $\text{Prob}(E)$ is the probability that $E$ occurs. If $E_1$ and $E_2$ are events, $\text{Prob}(E_1 \,|\, E_2)$ is the conditional probability that $E_1$ occurs given that $E_2$ occurs. The model for transmitting binary data we explore is called the **binary symmetric channel (BSC) with crossover probability** $\varrho$ as illustrated in Figure 1.2. In a BSC, we have the following conditional probabilities: For $y, c \in \mathbb{F}_2$,

$$\text{Prob}(y \text{ is received} \,|\, c \text{ is sent}) = \begin{cases} 1-\varrho & \text{if } y = c, \\ \varrho & \text{if } y \neq c. \end{cases} \tag{1.4}$$

In a BSC we also assume that the probability of error in one bit is independent of previous bits. We will assume that it is more likely that a bit is received correctly than in error; so $\varrho < \frac{1}{2}$. [10]

Let $\mathcal{C}$ be a binary code of length $n$. Assume that $\mathbf{c} \in \mathcal{C}$ is sent and $\mathbf{y} \in \mathbb{F}_2^n$ is received and decoded as $\widetilde{\mathbf{c}} \in \mathcal{C}$. Of course the hope is that $\widetilde{\mathbf{c}} = \mathbf{c}$; otherwise the decoder has made a **decoding error**. So $\text{Prob}(\mathbf{c} \,|\, \mathbf{y})$ is the probability that the codeword $\mathbf{c}$ is sent given that $\mathbf{y}$ is received, and $\text{Prob}(\mathbf{y} \,|\, \mathbf{c})$ is the probability that $\mathbf{y}$ is received given that the codeword $\mathbf{c}$

[10] While $\varrho$ is usually very small, if $\varrho > \frac{1}{2}$, the probability that a bit is received in error is higher than the probability that it is received correctly. One strategy is to then immediately interchange 0 and 1 at the receiving end. This converts the BSC with crossover probability $\varrho$ to a BSC with crossover probability $1 - \varrho < \frac{1}{2}$. This of course does not help if $\varrho = \frac{1}{2}$; in this case communication is not possible.

is sent. These probabilities are related by Bayes' Rule

$$\text{Prob}(\mathbf{c}\,|\,\mathbf{y}) = \frac{\text{Prob}(\mathbf{y}\,|\,\mathbf{c})\text{Prob}(\mathbf{c})}{\text{Prob}(\mathbf{y})}$$

where $\text{Prob}(\mathbf{c})$ is the probability that $\mathbf{c}$ is sent and $\text{Prob}(\mathbf{y})$ is the probability that $\mathbf{y}$ is received. There are two natural means by which a decoder can make a choice based on these two probabilities. First the decoder could decode $\mathbf{y}$ as $\widetilde{\mathbf{c}} \in \mathcal{C}$ where $\text{Prob}(\widetilde{\mathbf{c}}\,|\,\mathbf{y})$ is maximum; such a decoder is called a **maximum *a posteriori* probability (MAP) decoder**. Symbolically a MAP decoder makes the decision

$$\widetilde{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} \text{Prob}(\mathbf{c}\,|\,\mathbf{y}).$$

Here $\arg\max_{\mathbf{c}\in\mathcal{C}} \text{Prob}(\mathbf{c}\,|\,\mathbf{y})$ is the argument $\mathbf{c}$ of the probability function $\text{Prob}(\mathbf{c}\,|\,\mathbf{y})$ that maximizes this probability. Alternately the decoder could decode $\mathbf{y}$ as $\widetilde{\mathbf{c}} \in \mathcal{C}$ where $\text{Prob}(\mathbf{y}\,|\,\widetilde{\mathbf{c}})$ is maximum; such a decoder is called a **maximum likelihood (ML) decoder**. Symbolically an ML decoder makes the decision

$$\widetilde{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} \text{Prob}(\mathbf{y}\,|\,\mathbf{c}).$$

We further analyze ML decoding over a BSC. If $\mathbf{y} = y_1y_2\cdots y_n$ and $\mathbf{c} = c_1c_2\cdots c_n$,

$$\text{Prob}(\mathbf{y}\,|\,\mathbf{c}) = \prod_{i=1}^{n} \text{Prob}(y_i\,|\,c_i)$$

since bit errors are independent. By (1.4) $\text{Prob}(y_i\,|\,c_i) = \varrho$ if $y_i \neq c_i$ and $\text{Prob}(y_i\,|\,c_i) = 1-\varrho$ if $y_i = c_i$. Therefore

$$\text{Prob}(\mathbf{y}\,|\,\mathbf{c}) = \varrho^{\text{d}_\text{H}(\mathbf{y},\mathbf{c})}(1-\varrho)^{n-\text{d}_\text{H}(\mathbf{y},\mathbf{c})} = (1-\varrho)^n \left(\frac{\varrho}{1-\varrho}\right)^{\text{d}_\text{H}(\mathbf{y},\mathbf{c})}. \tag{1.5}$$

Since $0 < \varrho < \frac{1}{2}$, $0 < \frac{\varrho}{1-\varrho} < 1$. Thus maximizing $\text{Prob}(\mathbf{y}\,|\,\mathbf{c})$ is equivalent to minimizing $\text{d}_\text{H}(\mathbf{y},\mathbf{c})$; so a ML decoder finds the codeword $\mathbf{c}$ closest to the received vector $\mathbf{y}$ in Hamming distance.

**Definition 1.17.2** If a decoder decodes a received vector $\mathbf{y}$ as the codeword $\mathbf{c}$ with $\text{d}_\text{H}(\mathbf{y},\mathbf{c})$ minimized, the decoder is called a **nearest neighbor decoder**.

From this discussion, on a BSC, maximum likelihood and nearest neighbor decoding are the same. We can certainly perform nearest neighbor decoding on any code over any field.

Before presenting an example of a nearest neighbor decoder, we need to establish the relationship between the minimum distance of a code and the error-correcting capability of the code under nearest neighbor decoding. Notice this theorem is valid for any code, linear or not, over any finite field.

**Theorem 1.17.3** *Let $\mathcal{C}$ be an $(n, M, d)_q$ code and $t = \left\lfloor \frac{d-1}{2} \right\rfloor$. If a codeword $\mathbf{c} \in \mathcal{C}$ is sent and $\mathbf{y}$ is received where $t$ or fewer errors have occurred, then $\mathbf{c}$ is the unique codeword closest to $\mathbf{y}$. In particular nearest neighbor decoding uniquely and correctly decodes any received vector in which at most $t$ errors have occurred in transmission.*

**Proof:** By definition $\mathbf{y} \in S_{q,n,t}(\mathbf{c})$, the sphere of radius $t$ in $\mathbb{F}_q^n$ centered at $\mathbf{c}$. By Theorem 1.9.5(b), spheres of radius $t$ centered at codewords are pairwise disjoint; hence if $\mathbf{y} \in S_{q,n,t}(\mathbf{c}_1)$ with $\mathbf{c}_1 \in \mathcal{C}$, then $\mathbf{c} = \mathbf{c}_1$. □

**Definition 1.17.4** A code $\mathcal{C}$ is a **$t$-error-correcting code** provided that whenever any $\mathbf{c} \in \mathcal{C}$ is transmitted and $\mathbf{y} \in \mathbb{F}_q^n$ is received, where $\mathbf{y}$ differs from $\mathbf{c}$ in at most $t$ coordinates, then every other codeword in $\mathcal{C}$ differs from $\mathbf{y}$ in more than $t$ coordinates.

**Remark 1.17.5** By Theorem 1.17.3, an $(n, M, d)_q$ code $\mathcal{C}$ is $t$-error-correcting for any $t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. Furthermore, when $M > 1$ and $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, there exist two distinct codewords such that the spheres of radius $t+1$ about them are not disjoint; if this were not the case, the minimum distance of $\mathcal{C}$ is in fact larger than $d$. Thus when $M > 1$ and $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, $\mathcal{C}$ is not $(t+1)$-error-correcting.[11]

The nearest neighbor decoding problem for an $(n, M, d)_q$ code becomes one of finding an efficient algorithm that will correct up to $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors. An obvious decoding algorithm is to examine all codewords until one is found with distance $t$ or less from the received vector. This is a realistic decoding algorithm only for $M$ small. Another obvious algorithm is to make a table consisting of a nearest codeword for each of the $q^n$ vectors in $\mathbb{F}_q^n$ and then look up a received vector in the table to decode it. This is impractical if $q^n$ is very large.

For an $[n, k, d]_q$ linear code, we can devise an algorithm using a table with $q^{n-k}$ rather than $q^n$ entries where one can find the nearest codeword by looking up one of these $q^{n-k}$ entries. This general nearest neighbor decoding algorithm for linear codes is called *syndrome decoding*, which is the subject of the remainder of the section.

**Definition 1.17.6** Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. For $\mathbf{y} \in \mathbb{F}_q^n$, the **coset of $\mathcal{C}$** with **coset representative $\mathbf{y}$** is $\mathbf{y} + \mathcal{C} = \{\mathbf{y} + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}\}$. The **weight** of the coset $\mathbf{y} + \mathcal{C}$ is the smallest weight of a vector in the coset, and any vector of this smallest weight in the coset is called a **coset leader**.

The next result follows from the theory of finite groups as a linear code is a group under addition.

**Theorem 1.17.7** *Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. The following hold for $\mathbf{y}, \mathbf{y}', \mathbf{e} \in \mathbb{F}_q^n$.*

(a) $\mathbf{y} + \mathcal{C} = \mathbf{y}' + \mathcal{C}$ *if and only if* $\mathbf{y} - \mathbf{y}' \in \mathcal{C}$.

(b) *Cosets of $\mathcal{C}$ all have size $q^k$.*

(c) *Cosets of $\mathcal{C}$ are either equal or disjoint. There are $q^{n-k}$ distinct cosets of $\mathcal{C}$ and they partition $\mathbb{F}_q^n$.*

(d) *If $\mathbf{e}$ is a coset representative of $\mathbf{y} + \mathcal{C}$, then $\mathbf{e} + \mathcal{C} = \mathbf{y} + \mathcal{C}$. In particular, if $\mathbf{e}$ is a coset leader of $\mathbf{y} + \mathcal{C}$, then $\mathbf{e} + \mathcal{C} = \mathbf{y} + \mathcal{C}$.*

(e) *Any coset of weight at most $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ has a unique coset leader.*

Let $\mathcal{C}$ be an $[n, k, d]_q$ code; fix a parity check matrix $H$ of $\mathcal{C}$. For $\mathbf{y} \in \mathbb{F}_q^n$, $\mathrm{syn}(\mathbf{y}) = H\mathbf{y}^\mathsf{T}$ is called the **syndrome of $\mathbf{y}$**. Syndromes are column vectors in $\mathbb{F}_q^{n-k}$. The code $\mathcal{C}$ consists of all vectors whose syndrome equals $\mathbf{0}^\mathsf{T}$. As $H$ has rank $n-k$, every column vector in $\mathbb{F}_q^{n-k}$ is a syndrome. From Theorem 1.17.7, if $\mathbf{y}, \mathbf{y}' \in \mathbb{F}_q^n$ are in the same coset of $\mathcal{C}$, then $\mathbf{y} - \mathbf{y}' = \mathbf{c} \in \mathcal{C}$. Therefore $\mathrm{syn}(\mathbf{y}) = H\mathbf{y}^\mathsf{T} = H(\mathbf{y}' + \mathbf{c})^\mathsf{T} = H\mathbf{y}'^\mathsf{T} + H\mathbf{c}^\mathsf{T} = H\mathbf{y}'^\mathsf{T} + \mathbf{0}^\mathsf{T} = \mathrm{syn}(\mathbf{y}')$. Conversely, if $\mathrm{syn}(\mathbf{y}) = \mathrm{syn}(\mathbf{y}')$, then $H(\mathbf{y} - \mathbf{y}')^\mathsf{T} = \mathbf{0}^\mathsf{T}$ and so $\mathbf{y} - \mathbf{y}' \in \mathcal{C}$. Thus we have the following theorem.

[11] In the trivial case where $M = 1$, $\mathcal{C}$ is $n$-error-correcting as every received vector decodes to the only codeword in $\mathcal{C}$. However, since the information rate (Definition 1.9.26 ) of such a code is 0, it is never used in practice.

**Theorem 1.17.8** *Two vectors belong to the same coset if and only if they have the same syndrome.*

Hence there is a one-to-one correspondence between cosets of $\mathcal{C}$ and syndromes. For $\mathbf{s} \in \mathbb{F}_q^{n-k}$, denote by $\mathcal{C}_\mathbf{s}$ the coset of $\mathcal{C}$ consisting of all vectors in $\mathbb{F}_q^n$ with syndrome $\mathbf{s}^\mathsf{T}$. Also let $\mathbf{e}_\mathbf{s}$ be a coset leader of $\mathcal{C}_\mathbf{s}$. Thus $\mathcal{C}_\mathbf{s} = \mathbf{e}_\mathbf{s} + \mathcal{C}$.

Suppose a codeword sent over a communication channel is received as a vector $\mathbf{y}$. In nearest neighbor decoding we seek a vector $\mathbf{e}$ of smallest weight such that $\mathbf{y} - \mathbf{e} \in \mathcal{C}$. So nearest neighbor decoding is equivalent to finding a coset leader $\mathbf{e}$ of the coset $\mathbf{y} + \mathcal{C}$ and decoding the received vector $\mathbf{y}$ as $\mathbf{y} - \mathbf{e}$. The *Syndrome Decoding Algorithm* is the following implementation of nearest neighbor decoding.

**Algorithm 1.17.9 (Syndrome Decoding)**

---

Use the above notation.

Step 1: For each syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$, choose a coset leader $\mathbf{e}_\mathbf{s}$ of the coset $\mathcal{C}_\mathbf{s}$. Create a table pairing the syndrome with the coset leader.

Step 2: After receiving a vector $\mathbf{y}$, compute $\mathbf{s} = \text{syn}(\mathbf{y})$.

Step 3: Decode $\mathbf{y}$ as the codeword $\mathbf{y} - \mathbf{e}_\mathbf{s}$.

---

Step 1 of this algorithm can be somewhat involved, but it is a one-time preprocessing task that is carried out before received vectors are analyzed. We briefly describe this table creation. Begin with all vectors in $\mathbb{F}_q^n$ of weight $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ or less and place them in the table paired with their syndromes; by Theorem 1.17.7(e), no syndrome is repeated. If all syndromes have not been accounted for, place all vectors in $\mathbb{F}_q^n$ of weight $t+1$, one at a time, paired with their syndromes into the table as long as the syndrome is not already in the table. If all syndromes have still not been accounted for, repeat this procedure with vectors in $\mathbb{F}_q^n$ of weight $t+2$, then weight $t+3$, and continue inductively. End the process when all syndromes are in the table.

**Example 1.17.10** Let $\mathcal{C}$ be the $[6,3,3]_2$ binary code with parity check matrix

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

The table of Step 1 in the Syndrome Decoding Algorithm is the following.

| leader | syndrome | leader | syndrome | leader | syndrome | leader | syndrome |
|---|---|---|---|---|---|---|---|
| 000000 | $000^\mathsf{T}$ | 010000 | $101^\mathsf{T}$ | 000100 | $100^\mathsf{T}$ | 000001 | $001^\mathsf{T}$ |
| 100000 | $011^\mathsf{T}$ | 001000 | $110^\mathsf{T}$ | 000010 | $010^\mathsf{T}$ | 100100 | $111^\mathsf{T}$ |

Notice that the coset with syndrome $111^\mathsf{T}$ has weight 2 and does not have a unique coset leader. This coset has two other coset leaders: 010010 and 001001. All other cosets have unique coset leaders by Theorem 1.17.7(e). We analyze three received vectors.

- Suppose $\mathbf{y} = 110110$ is received. Then $\text{syn}(\mathbf{y}) = H\mathbf{y}^\mathsf{T} = 000^\mathsf{T}$ and $\mathbf{y}$ is decoded as $\mathbf{y}$. $\mathbf{y}$ was the sent codeword provided 2 or more errors were not made.

- Now suppose that $\mathbf{y} = 101000$ is received. Then $\text{syn}(\mathbf{y}) = 101^{\mathsf{T}}$ and $\mathbf{y}$ is decoded as $\mathbf{y} - 010000 = 111000$. This was the sent codeword provided only 1 error was made.
- Finally suppose that $\mathbf{y} = 111111$ is received. Then $\text{syn}(\mathbf{y}) = 111^{\mathsf{T}}$ and $\mathbf{y}$ is decoded as $\mathbf{y} - 100100 = 011011$ and at least 2 errors were made in transmission. If exactly 2 errors were made, and we had chosen one of the other two possible coset leaders for the table, $\mathbf{y}$ would have been decoded as $\mathbf{y} - 010010 = 101101$ or $\mathbf{y} - 001001 = 110110$.

For this code, any received vector where 0 or 1 errors were made would be decoded correctly. If 2 errors were made, the decoder would decode the received vector to one of three possible equally likely codewords; there is no way to determine which was actually sent. If more than 2 errors were made, the decoder would always decode the received vector incorrectly.

**Example 1.17.11** Nearest neighbor decoding of the binary Hamming code $\mathcal{H}_{m,2}$ is particularly easy. The parity check matrix for this code consists of the $2^m - 1$ nonzero binary $m$-tuples of column length $m$; these can be viewed as the binary expansions of the integers $1, 2, \ldots, 2^m - 1$. Choose the parity check matrix $H$ for $\mathcal{H}_{m,2}$ where column $i$ is the associated binary $m$-tuple expansion of $i$. Step 1 of the Syndrome Decoding Algorithm 1.17.9 can be skipped and the algorithm becomes the following: If $\mathbf{y}$ is received, compute $\mathbf{s} = \text{syn}(\mathbf{y})$. If $\mathbf{s} = \mathbf{0}^{\mathsf{T}}$, decode $\mathbf{y}$ as the codeword $\mathbf{y}$. Otherwise $\mathbf{s}$ represents the binary expansion of some integer $i$; the nearest codeword $\mathbf{c}$ to $\mathbf{y}$ is obtained from $\mathbf{y}$ by adding 1 to the $i^{\text{th}}$ bit.

As an illustration, the parity check matrix to use for $\mathcal{H}_{4,2}$ is

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

Suppose $\mathbf{y} = 100110001111000$ is received. Then $\text{syn}(\mathbf{y}) = 0100^{\mathsf{T}}$, which is column 4 of $H$. Hence 1 is added to coordinate 4 of $\mathbf{y}$ to yield the codeword $\mathbf{c} = 100010001111000$.

## 1.18 Shannon's Theorem

Shannon's Channel Coding Theorem [1661] guarantees that good codes exist making reliable communication possible. We will discuss this theorem in the context of binary linear codes for which maximum likelihood decoding over a BSC is used. Note however that the theorem can be stated in a more general setting.

Assume that the communication channel is a BSC with crossover probability $\varrho$ and that syndrome decoding is used as the implementation of ML decoding to decode an $[n, k, d]_2$ code $\mathcal{C}$. The **word error rate** $\text{P}_{\text{err}}$ for this channel and decoding scheme is the probability that the decoder makes an error, averaged over all codewords of $\mathcal{C}$; for simplicity assume that each codeword of $\mathcal{C}$ is equally likely to be sent. A decoder error occurs when $\widetilde{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} \text{Prob}(\mathbf{y} \,|\, \mathbf{c})$ is not the originally transmitted codeword $\mathbf{c}$ when $\mathbf{y}$ is received. The syndrome decoder makes a correct decision if $\mathbf{y} - \mathbf{c}$ is a coset leader. The probability that the decoder makes a correct decision is

$$\varrho^{\text{wt}_{\text{H}}(\mathbf{y}-\mathbf{c})}(1-\varrho)^{n-\text{wt}_{\text{H}}(\mathbf{y}-\mathbf{c})}$$

by (1.5). Therefore the probability that the syndrome decoder makes a correct decision

averaged over all equally likely transmitted codewords is $\sum_{i=0}^{n} \alpha_i \varrho^i (1-\varrho)^{n-i}$ where $\alpha_i$ is the number of coset leaders of weight $i$. Thus

$$\mathrm{P}_{\mathrm{err}} = 1 - \sum_{i=0}^{n} \alpha_i \varrho^i (1-\varrho)^{n-i}. \tag{1.6}$$

**Example 1.18.1** Suppose binary messages of length $k$ are sent unencoded over a BSC with crossover probability $\varrho$. This in effect is the same as transmitting codewords from the $[k, k, 1]_2$ code $\mathcal{C} = \mathbb{F}_2^k$. This code has a unique coset, the code itself, and its leader is the zero codeword of weight 0. Hence $\alpha_0 = 1$ and $\alpha_i = 0$ for $i > 0$. Therefore (1.6) shows that the probability of decoder error is

$$\mathrm{P}_{\mathrm{err}} = 1 - \varrho^0 (1-\varrho)^k = 1 - (1-\varrho)^k.$$

This is precisely what we expect as the probability of no decoding error is the probability $(1-\varrho)^k$ that the $k$ bits are received without error. For instance if $\varrho = 0.01$ and $k = 4$, $\mathrm{P}_{\mathrm{err}}$ without coding the length 4 messages is 0.03940399.

**Example 1.18.2** We compare sending $2^4 = 16$ binary messages unencoded to encoding using the $[7, 4, 3]_2$ binary Hamming code $\mathcal{H}_{3,2}$. By Theorem 1.17.7(c), there are $2^{7-4} = 8$ cosets of $\mathcal{H}_{3,2}$ in $\mathbb{F}_2^7$. Since $\mathbb{F}_2^7$ has 1 vector of weight 0 and 7 vectors of weight 1, these must be the coset leaders for all 8 cosets of $\mathcal{H}_{3,2}$ in $\mathbb{F}_2^7$ by Theorem 1.17.7(e). Thus $\alpha_0 = 1$, $\alpha_1 = 7$, and $\alpha_i = 0$ for $i > 1$. Hence the probability of decoder error is

$$\mathrm{P}_{\mathrm{err}} = 1 - (1-\varrho)^7 - 7\varrho(1-\varrho)^6$$

by (1.6). For example if $\varrho = 0.01$, $\mathrm{P}_{\mathrm{err}} = 0.00203104\cdots$, significantly lower than the word error rate for unencoded transmissions of binary messages of length 4 found in Example 1.18.1. For comparison, when transmitting 10 000 unencoded binary messages each of length 4, one can expect about 394 to be received in error. On the other hand, when transmitting 10 000 binary messages each of length 4 encoded to length 7 codewords from $\mathcal{H}_{3,2}$, one can expect about 20 to be decoded in error.

In order to state Shannon's Theorem, we need to define the channel capacity.

**Definition 1.18.3** For a BSC with crossover probability $\varrho$, the **capacity of the channel** is

$$C(\varrho) = 1 + \varrho \log_2 \varrho + (1-\varrho)\log_2(1-\varrho).$$

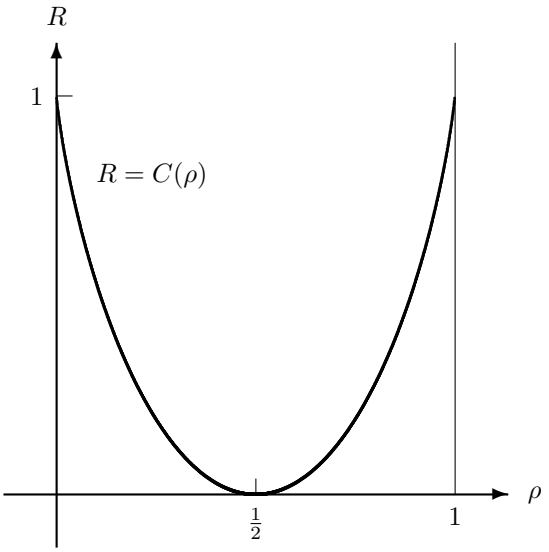
The capacity $C(\varrho) = 1 - H_2(\varrho)$ where $H_2(\varrho)$ is the binary entropy function defined in more generality in Section 1.9.8.[12] See Figure 1.3.

The next theorem is Shannon's Theorem for binary symmetric channels. Shannon's original theorem was stated for nonlinear codes but was later shown to be valid for linear codes as well. The theorem also holds for other channels provided channel capacity is appropriately defined. For discussion and proofs of various versions of Shannon's Theorem, see for example [467, 1314]. For binary symmetric channels, Shannon's Theorem is as follows.

**Theorem 1.18.4 (Shannon)** *Let $\delta > 0$ and $R < C(\varrho)$. Then for large enough $n$, there exists an $[n, k]_2$ binary linear code $\mathcal{C}$ with $\frac{k}{n} \geq R$ such that $\mathrm{P}_{\mathrm{err}} < \delta$ when $\mathcal{C}$ is used for communication over a BSC with crossover probability $\varrho$. Furthermore no such code exists if $R > C(\varrho)$.*

[12] When $q = 2$, the domain of the entropy function $H_2(x)$ can be extended from $0 \leq x < \frac{1}{2}$ to $0 \leq x < 1$.

**FIGURE 1.3**: Channel capacity for a BCS with crossover probability $\rho$




**Remark 1.18.5** When the crossover probability is $\rho = \frac{1}{2}$, $C\left(\frac{1}{2}\right) = 0$. In this case Shannon's Theorem indicates that communication is not possible. This is not surprising; when $\rho = \frac{1}{2}$, whether a binary symbol is received correctly or incorrectly is essentially determined by a coin flip. See Footnote 10.

**Remark 1.18.6** Recall that $\frac{k}{n}$ is the information rate of the code as in Definition 1.9.26. The proof of Shannon's Theorem is nonconstructive, but the theorem does guarantee that good codes exist with information rates just under channel capacity and decoding error rates arbitrarily small; unfortunately these codes may have to be extremely long. The objective becomes to find codes with a large number of codewords (to send many messages), large minimum distance (to correct many errors), and short length (to minimize transmission time or storage space). These goals conflict as seen in Section 1.9.

# Chapter 2

# Cyclic Codes over Finite Fields

**Cunsheng Ding**

*The Hong Kong University of Science and Technology*



## 2.1 Notation and Introduction

A brief introduction to cyclic codes over finite fields was given in Section 1.12. The objective of this chapter is to introduce several important families of cyclic codes over finite fields. We will follow the notation of Chapter 1 as closely as possible.

By an $[n, \kappa, d]_q$ code, we mean a linear code over $\mathbb{F}_q$ with length $n$, dimension $\kappa$ and minimum distance $d$. Notice that the minimum distance of a linear code is equal to the minimum nonzero weight of the code. By the parameters of a linear code, we mean its length, dimension and minimum distance. An $[n, \kappa, d]_q$ code is said to be **distance-optimal** (respectively **dimension-optimal**) if there is no $[n, \kappa, d+1]_q$ (respectively $[n, \kappa+1, d]_q$) code. By the best known parameters of $[n, \kappa]$ linear codes over $\mathbb{F}_q$ we mean an $[n, \kappa, d]_q$ code with the largest known $d$ reported in the tables of linear codes maintained at [845].

In this chapter, we deal with cyclic codes of length $n$ over $\mathbb{F}_q$ and always assume that $\gcd(n, q) = 1$. Under this assumption, $x^n - 1$ has no repeated factors over $\mathbb{F}_q$. Denote by $C_i$ the $q$-cyclotomic coset modulo $n$ that contains $i$ for $0 \le i \le n-1$. Put $m = \mathrm{ord}_n(q)$, and let $\gamma$ be a generator of $\mathbb{F}_{q^m}^* := \mathbb{F}_{q^m} \setminus \{0\}$. Define $\alpha = \gamma^{(q^m-1)/n}$. Then $\alpha$ is a primitive $n^{\text{th}}$ root of unity. The canonical factorization of $x^n - 1$ over $\mathbb{F}_q$ is given by

$$x^n - 1 = M_{\alpha^{i_0}}(x) M_{\alpha^{i_1}}(x) \cdots M_{\alpha^{i_t}}(x), \tag{2.1}$$

where $i_0, i_1, \ldots, i_t$ are representatives of the $q$-cyclotomic cosets modulo $n$, and

$$M_{\alpha^{i_j}}(x) = \prod_{h \in C_{i_j}} (x - \alpha^h),$$

which is the minimal polynomial of $\alpha^{i_j}$ over $\mathbb{F}_q$ and is irreducible over $\mathbb{F}_q$.

Throughout this chapter, we define $\mathcal{R}_{(n,q)} = \mathbb{F}_q[x]/\langle x^n - 1\rangle$ and use $\mathrm{Tr}_{q^m/q}$ to denote the trace function from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$ defined by $\mathrm{Tr}_{q^m/q}(x) = \sum_{j=0}^{m-1} x^{q^j}$. The ring of integers modulo $n$ is denoted by $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$.

Cyclic codes form an important subclass of linear codes over finite fields. Their algebraic structure is richer. Because of their cyclic structure, they are closely related to number theory. In addition, they have efficient encoding and decoding algorithms and are the most studied linear codes. In fact, most of the important families of linear codes are either cyclic codes or extended cyclic codes.

---

## 2.2 Subfield Subcodes

Let $\mathcal{C}$ be an $[n, \kappa]_{q^t}$ code. The **subfield subcode** $\mathcal{C}|_{\mathbb{F}_q}$ of $\mathcal{C}$ with respect to $\mathbb{F}_q$ is the set of codewords in $\mathcal{C}$ each of whose components is in $\mathbb{F}_q$. Since $\mathcal{C}$ is linear over $\mathbb{F}_{q^t}$, $\mathcal{C}|_{\mathbb{F}_q}$ is a linear code over $\mathbb{F}_q$.

The dimension, denoted $\kappa_q$, of the subfield subcode $\mathcal{C}|_{\mathbb{F}_q}$ may not have an elementary relation with that of the code $\mathcal{C}$. However, we have the following lower and upper bounds on $\kappa_q$.

**Theorem 2.2.1** *Let $\mathcal{C}$ be an $[n, \kappa]_{q^t}$ code. Then $\mathcal{C}|_{\mathbb{F}_q}$ is an $[n, \kappa_q]$ code over $\mathbb{F}_q$, where $\kappa \geq \kappa_q \geq n - t(n - \kappa)$. If $\mathcal{C}$ has a basis of codewords in $\mathbb{F}_q^n$, then this is also a basis of $\mathcal{C}|_{\mathbb{F}_q}$ and $\mathcal{C}|_{\mathbb{F}_q}$ has dimension $\kappa$.*

**Example 2.2.2** The Hamming code $\mathcal{H}_{3,2^2}$ over $\mathbb{F}_{2^2}$ has parameters $[21, 18, 3]_4$. The subfield subcode $\mathcal{H}_{3,2^2}|_{\mathbb{F}_2}$ is a $[21, 16, 3]_2$ code with parity check matrix

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

In this case, $n = 21$, $\kappa = 18$, and $n - t(n - \kappa) = 15$. Hence $\kappa_q = 16$, which is very close to $n - t(n - \kappa) = 15$.

The following is called **Delsarte's Theorem**, which exhibits a dual relation between subfield subcodes and trace codes. This theorem is very useful in the design and analysis of linear codes.

**Theorem 2.2.3 (Delsarte)** *Let $\mathcal{C}$ be a linear code of length $n$ over $\mathbb{F}_{q^t}$. Then*

$$(\mathcal{C}|_{\mathbb{F}_q})^\perp = \mathrm{Tr}_{q^t/q}(\mathcal{C}^\perp),$$

*where* $\mathrm{Tr}_{q^t/q}(\mathcal{C}^\perp) = \left\{ \left(\mathrm{Tr}_{q^t/q}(v_1), \ldots, \mathrm{Tr}_{q^t/q}(v_n)\right) \mid (v_1, \ldots, v_n) \in \mathcal{C}^\perp \right\}$.

Theorems 2.2.1 and 2.2.3 work for all linear codes, including cyclic codes. Their proofs could be found in [1008, Section 3.8]. We shall need them later.

## 2.3 Fundamental Constructions of Cyclic Codes

In Section 1.12, it was shown that every cyclic code of length $n$ over $\mathbb{F}_q$ can be generated by a generator polynomial $g(x) \in \mathbb{F}_q[x]$. The objective of this section is to describe several other fundamental constructions of cyclic codes over finite fields. By a fundamental construction, we mean a construction method that can produce every cyclic code over any finite field.

An element $e$ in a commutative ring $\mathcal{R}$ is called an **idempotent** if $e^2 = e$. The ring $\mathcal{R}_{(n,q)}$ has in general quite a number of idempotents. Besides its generator polynomial, many other polynomials can generate a cyclic code $\mathcal{C}$. Let $\mathcal{C}$ be a cyclic code over $\mathbb{F}_q$ with generator polynomial $g(x)$. It is easily seen that a polynomial $f(x) \in \mathbb{F}_q[x]$ generates $\mathcal{C}$ if and only if $\gcd(f(x), x^n - 1) = g(x)$.

If an idempotent $e(x) \in \mathcal{R}_{(n,q)}$ generates a cyclic code $\mathcal{C}$, it is then unique in this ring and called the **generating idempotent**. Given the generator polynomial of a cyclic code, one can compute its generating idempotent with the following theorem [1008, Theorem 4.3.3].

**Theorem 2.3.1** *Let $\mathcal{C}$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with generator polynomial $g(x)$. Let $h(x) = (x^n - 1)/g(x)$. Then $\gcd(g(x), h(x)) = 1$, as it was assumed that $\gcd(n, q) = 1$. Employing the Extended Euclidean Algorithm, one computes two polynomials $a(x) \in \mathbb{F}_q[x]$ and $b(x) \in \mathbb{F}_q[x]$ such that $1 = a(x)g(x) + b(x)h(x)$. Then $e(x) = a(x)g(x) \bmod (x^n - 1)$ is the generating idempotent of $\mathcal{C}$.*

The polynomial $h(x)$ in Theorem 2.3.1 is called the **parity check polynomial** of $\mathcal{C}$. Given the generating idempotent of a cyclic code, one obtains the generator polynomial of this code as follows [1008, Theorem 4.3.3].

**Theorem 2.3.2** *Let $\mathcal{C}$ be a cyclic code over $\mathbb{F}_q$ with generating idempotent $e(x)$. Then the generator polynomial of $\mathcal{C}$ is given by $g(x) = \gcd(e(x), x^n - 1)$, which is computed in $\mathbb{F}_q[x]$.*

**Example 2.3.3** The cyclic code $\mathcal{C}$ of length 11 over $\mathbb{F}_3$ with generator polynomial $g(x) = x^5 + x^4 + 2x^3 + x^2 + 2$ has parameters $[11, 6, 5]$ and parity check polynomial $h(x) = x^6 + 2x^5 + 2x^4 + 2x^3 + x^2 + 1$.

Let $a(x) = 2x^5 + x^4 + x^2$ and $b(x) = x^4 + x^3 + 1$. It is then easily verified that $1 = a(x)g(x) + b(x)h(x)$. Hence

$$e(x) = a(x)g(x) \bmod (x^{11} - 1) = 2x^{10} + 2x^8 + 2x^7 + 2x^6 + 2x^2,$$

which is the generating idempotent of $\mathcal{C}$. On the other hand, we have $g(x) = \gcd(e(x), x^{11} - 1)$.

A generator matrix of a cyclic code can be derived from its generating idempotent as follows [1008, Theorem 4.3.6].

**Theorem 2.3.4** *If $\mathcal{C}$ is an $[n, \kappa]$ cyclic code with generating idempotent $e(x) = \sum_{i=0}^{n-1} e_i x^i$, then the $\kappa \times n$ matrix*

$$\begin{bmatrix} e_0 & e_1 & e_2 & \cdots & e_{n-2} & e_{n-1} \\ e_{n-1} & e_0 & e_1 & \cdots & e_{n-3} & e_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{n-\kappa+1} & e_{n-\kappa+2} & e_{n-\kappa+3} & \cdots & e_{n-\kappa-1} & e_{n-\kappa} \end{bmatrix}$$

*is a generator matrix of $\mathcal{C}$.*

Let $f(x) = f_0 + f_1 x + \cdots + f_\ell x^\ell$ be a polynomial over a field with $f_\ell \neq 0$. Then the **reciprocal** of $f$ is defined by

$$f^\perp(x) = f_\ell^{-1} x^\ell f(x^{-1}).$$

Another fundamental construction of cyclic codes over finite fields is the following trace construction [1899], which is a direct consequence of Theorem 2.2.3.

**Theorem 2.3.5** *Let $\mathcal{C}$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with parity check polynomial $h(x)$. Let $J$ be a subset of $\mathbb{Z}_n$ such that*

$$h^\perp(x) = \prod_{j \in J} M_{\alpha^j}(x),$$

*where $h^\perp(x)$ is the reciprocal of $h(x)$. Then $\mathcal{C}$ consists of all the following codewords:*

$$c_a(x) = \sum_{i=0}^{n-1} \mathrm{Tr}_{q^m/q}(f_a(\alpha^i))x^i$$

*where*

$$f_a(x) = \sum_{j \in J} a_j x^j \quad \text{for } a_j \in \mathbb{F}_{q^m}.$$

The trace and generator polynomial approaches are the most popular and effective ways for constructing and analysing cyclic codes. In particular, the trace approach allows the use of exponential sums for the determination of the weight distribution of cyclic codes. A lot of progress in this direction has been made in the past decade [560]. A less investigated fundamental approach to cyclic codes is the $q$-polynomial method developed in [562]. Another fundamental construction uses sequences as described in Section 20.5.

The following theorem says that every projective linear code is a punctured code of a special cyclic code [1914].

**Theorem 2.3.6** *Every linear code $\mathcal{C}$ of length $n$ over $\mathbb{F}_q$ with minimum distance of $\mathcal{C}^\perp$ at least 3 is a punctured code of the cyclic code*

$$\left\{ \left( \mathrm{Tr}_{q^m/q}(a\gamma^0), \mathrm{Tr}_{q^m/q}(a\gamma^1), \ldots, \mathrm{Tr}_{q^m/q}(a\gamma^{q^m-2}) \right) \mid a \in \mathbb{F}_{q^m} \right\}.$$

*for some integer $m$, where $\gamma$ is a generator of $\mathbb{F}_{q^m}^*$.*

## 2.4 The Minimum Distances of Cyclic Codes

The length of a cyclic code is clear from its definition. However, determining the dimensions and minimum distances of cyclic codes is nontrivial. If a cyclic code $\mathcal{C}$ of length $n$ is defined by its generator polynomial $g(x)$, then the dimension of $\mathcal{C}$ equals $n - \deg(g)$. But it may be hard to find the degree of $g(x)$ when $g(x)$ is given as the least common multiple of a number of polynomials. If a cyclic code is defined in the trace form, it may also be difficult to determine the dimension. Determining the exact minimum distance of a cyclic code is more difficult. In the case that the minimum distance of a cyclic code cannot be settled, the best one could expect is to develop a good lower bound on the minimum distance. Unlike many other subclasses of linear codes, cyclic codes have some lower bounds on their

minimum distances. Some of the bounds are easy to use, while others are hard to employ. Below we introduce a few effective lower bounds on the minimum distances of cyclic codes.

Let $\mathcal{C}$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with generator polynomial

$$g(x) = \prod_{i \in T} (x - \alpha^i)$$

where $T$ is the union of some $q$-cyclotomic cosets modulo $n$, and is called the **defining set of $\mathcal{C}$ relative to** $\alpha$. The following is a simple but very useful lower bound ([248] and [968]).

**Theorem 2.4.1 (BCH Bound)** *Let $\mathcal{C}$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with defining set $T$ and minimum distance $d$. Assume $T$ contains $\delta - 1$ consecutive integers for some integer $\delta$. Then $d \geq \delta$.*

The BCH Bound depends on the choice of the primitive $n^{\text{th}}$ root of unity $\alpha$. Different choices of the primitive root may yield different lower bounds. When applying the BCH Bound, it is crucial to choose the right primitive root. However, it is open how to choose such a primitive root. In many cases the BCH Bound may be far away from the actual minimum distance. In such cases, the lower bound given in the following theorem may be much better. It was discovered by Hartmann and Tzeng [914]. To introduce this bound, we define

$$A + B = \{a + b \mid a \in A,\ b \in B\},$$

where $A$ and $B$ are two subsets of the ring $\mathbb{Z}_n$, $n$ is a positive integer, and $+$ denotes the integer addition modulo $n$.

**Theorem 2.4.2 (Hartmann–Tzeng Bound)** *Let $\mathcal{C}$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with defining set $T$ and minimum distance $d$. Let $A$ be a set of $\delta - 1$ consecutive elements of $T$ and $B(b, s) = \{jb \bmod n \mid 0 \leq j \leq s\}$, where $\gcd(b, n) < \delta$. If $A + B(b, s) \subseteq T$ for some $b$ and $s$, then $d \geq \delta + s$.*

When $s = 0$, the Hartmann–Tzeng Bound becomes the BCH Bound. Other lower bounds can be found in [1838]. As cyclic codes are a special case of quasi-cyclic codes, bounds on such codes found in Section 7.4 can be applied to cyclic codes.

---

## 2.5 Irreducible Cyclic Codes

Let $\mathcal{C}(q, n, i)$ denote the cyclic code of length $n$ over $\mathbb{F}_q$ with parity check polynomial $M_{\alpha^i}(x)$, which is the minimal polynomial of $\alpha^i$ over $\mathbb{F}_q$, and where $\alpha$ is a primitive $n^{\text{th}}$ root of unity over an extension field of $\mathbb{F}_q$. These $\mathcal{C}(q, n, i)$ are called **irreducible cyclic codes.** Since the ideals $\langle (x^n - 1)/M_{\alpha^i}(x) \rangle$ of $\mathcal{R}_{(n,q)}$ are minimal, these $\mathcal{C}(q, n, i)$ are also called **minimal cyclic codes.**

By Theorem 2.3.5, $\mathcal{C}(q, n, i)$ has the following trace representation:

$$\mathcal{C}(q, n, i) = \left\{ \left( \mathrm{Tr}_{q^{m_i}/q}(a\beta^0), \mathrm{Tr}_{q^{m_i}/q}(a\beta), \ldots, \mathrm{Tr}_{q^{m_i}/q}(a\beta^{n-1}) \right) \mid a \in \mathbb{F}_{q^{m_i}} \right\},$$

where $\beta = \alpha^{-i} \in \mathbb{F}_{q^{m_i}}$ and $m_i = |C_i|$.

**Example 2.5.1** Let $n = (q^m - 1)/(q - 1)$ and $\alpha = \gamma^{q-1}$, where $\gamma$ is a generator of $\mathbb{F}_{q^m}^*$. If $\gcd(q - 1, m) = 1$, then $\mathcal{C}(q, n, 1)$ has parameters $[n, m, q^{m-1}]$ and is equivalent to the simplex code whose dual is the Hamming code. Hence, when $\gcd(q-1, m) = 1$, the Hamming code is equivalent to a cyclic code.

**Example 2.5.2** The celebrated Golay codes introduced in Section 1.13 are also irreducible cyclic codes and the binary [24, 12, 8] extended Golay code was used on the Voyager 1 and Voyager 2 missions to Jupiter, Saturn, and their moons.

By definition, the dimension of $\mathcal{C}(q, n, i)$ equals $\deg(M_{\alpha^i}(x))$, which is a divisor of $m := \mathrm{ord}_n(q)$. The determination of the weight enumerators of irreducible cyclic codes is equivalent to the evaluation of Gaussian periods, which is extremely difficult in general. However, in a small number of cases, the weight enumerator of some irreducible cyclic codes is known. One-weight, two-weight and three-weight irreducible cyclic codes exist. It is in general hard to determine the minimum distance of an irreducible cyclic code. A lower bound on the minimum distances of irreducible cyclic codes has been developed. The reader is referred to [568] for detailed information.

Irreducible cyclic codes are very important for many reasons. First of all, every cyclic code is the direct sum of a number of irreducible cyclic codes. Secondly, the automorphism group of some irreducible codes (Golay codes) has high transitivity. Thirdly, some irreducible codes can be employed to construct maximal arcs, elliptic quadrics (ovoids), inversive planes, and $t$-designs. Hence, irreducible cyclic codes are closely related to group theory, finite geometry and combinatorics. In addition, irreducible cyclic codes also have a number of applications in engineering.

## 2.6 BCH Codes and Their Properties

BCH codes are a subclass of cyclic codes with special properties and are important in both theory and practice. Experimental data shows that binary and ternary BCH codes of certain lengths are the best cyclic codes in almost all cases; see [549, Appendix A]. BCH codes were briefly introduced in Section 1.14. This section treats BCH codes further and summarizes their basic properties.

Let $\delta$ be an integer with $2 \leq \delta \leq n$ and let $b$ be an integer. A **BCH code** over $\mathbb{F}_q$ of length $n$ and **designed distance** $\delta$, denoted by $\mathcal{C}_{(q,n,\delta,b)}$, is a cyclic code with defining set

$$T(b, \delta) = C_b \cup C_{b+1} \cup \cdots \cup C_{b+\delta-2} \tag{2.2}$$

relative to the primitive $n^{\text{th}}$ root of unity $\alpha$, where $C_i$ is the $q$-cyclotomic coset modulo $n$ containing $i$.

When $b = 1$, the code $\mathcal{C}_{(q,n,\delta,b)}$ with defining set in (2.2) is called a **narrow-sense** BCH code. If $n = q^m - 1$, then $\mathcal{C}_{(q,n,\delta,b)}$ is referred to as a **primitive** BCH code. The Reed–Solomon code introduced in Section 1.14 is a primitive BCH code.

Sometimes $T(b_1, \delta_1) = T(b_2, \delta_2)$ for two distinct pairs $(b_1, \delta_1)$ and $(b_2, \delta_2)$. The **maximum designed distance** of a BCH code is defined to be the largest $\delta$ such that the set $T(b, \delta)$ in (2.2) defines the code for some $b \geq 0$. The maximum designed distance of a BCH code is also called the **Bose distance**.

Given the canonical factorization of $x^n - 1$ over $\mathbb{F}_q$ in (2.1), we know that the total number of nonzero cyclic codes of length $n$ over $\mathbb{F}_q$ is $2^{t+1} - 1$. Then the following two natural questions arise:

1. How many of the $2^{t+1} - 1$ cyclic codes are BCH codes?
2. Which of the $2^{t+1} - 1$ cyclic codes are BCH codes?

The first question is open. Regarding the second question, we have the next result whose proof is straightforward.

**Theorem 2.6.1** *A cyclic code of length $n$ over $\mathbb{F}_q$ with defining set $T \subseteq \mathbb{Z}_n$ is a BCH code if and only if there exists an integer $\delta$ with $2 \leq \delta \leq n$, an integer $b$ with $-(n-1) \leq b \leq n-1$, and an element $a \in \mathbb{Z}_n$ such that $\gcd(n, a) = 1$ and*

$$aT \bmod n = \bigcup_{i=0}^{\delta-2} C_{i+b}.$$

In general it is not easy to use Theorem 2.6.1 to check if a cyclic code is a BCH code. In addition, it looks hard to answer the first question above with Theorem 2.6.1.

**Definition 2.6.2** A family of codes is **asymptotically good**, provided that there exists an infinite subset of $[n_i, k_i, d_i]$ codes from this family with $\lim_{i\to\infty} n_i = \infty$ such that both $\liminf_{i\to\infty} k_i/n_i > 0$ and $\liminf_{i\to\infty} d_i/n_i > 0$.

The family of primitive BCH codes over $\mathbb{F}_q$ is asymptotically bad in the following sense [1262].

**Theorem 2.6.3** *If $\mathcal{C}_i$ are $[n_i, k_i, d_i]_q$ codes for $i = 1, 2, \ldots$ with $\lim_{i\to\infty} n_i = \infty$, then either $\liminf_{i\to\infty} k_i/n_i = 0$ or $\liminf_{i\to\infty} d_i/n_i = 0$.*

Despite this asymptotic property, binary primitive BCH codes of length up to 127 are among the best linear codes known [549]. It is questionable if the definition of asymptotic badness above makes sense for applications.

### 2.6.1 The Minimum Distances of BCH Codes

It follows from Theorem 2.4.1 that a cyclic code with designed distance $\delta$ has minimum weight at least $\delta$. It is possible that the actual minimum distance is equal to the designed distance. Sometimes the actual minimum distance is much larger than the designed distance.

A codeword $(c_0, \ldots, c_{n-1})$ of a linear code $\mathcal{C}$ is **even-like** if $\sum_{j=0}^{n-1} c_j = 0$, and **odd-like** otherwise. The weight of an even-like (respectively odd-like) codeword is called an **even-like weight** (respectively **odd-like weight**). Let $\mathcal{C}$ be a primitive narrow-sense BCH code of length $n = q^m - 1$ over $\mathbb{F}_q$ with designed distance $\delta$. The defining set is then $T(1, \delta) = C_1 \cup C_2 \cup \cdots \cup C_{\delta-1}$. The following theorem provides useful information on the minimum weight of narrow-sense primitive BCH codes.

**Theorem 2.6.4** *Let $\mathcal{C}$ be the narrow-sense primitive BCH code of length $n = q^m - 1$ over $\mathbb{F}_q$ with designed distance $\delta$. Then the minimum weight of $\mathcal{C}$ is its minimum odd-like weight.*

The coordinates of the narrow-sense primitive BCH code $\mathcal{C}$ of length $n = q^m - 1$ over $\mathbb{F}_q$ with designed distance $\delta$ can be indexed by the elements of $\mathbb{F}_{q^m}^*$, and the extended coordinate in the extended code $\widehat{\mathcal{C}}$ can be indexed by the zero element of $\mathbb{F}_{q^m}$. The general affine group $\mathrm{GA}_1(\mathbb{F}_{q^m})$ then acts on $\mathbb{F}_{q^m}$ and also on $\widehat{\mathcal{C}}$ doubly transitively, where

$$\mathrm{GA}_1(\mathbb{F}_{q^m}) = \{ax + b \mid a \in \mathbb{F}_{q^m}^*,\ b \in \mathbb{F}_{q^m}\}.$$

Since $\mathrm{GA}_1(\mathbb{F}_{q^m})$ is transitive on $\mathbb{F}_{q^m}$, it is a subgroup of the permutation automorphism group of $\widehat{\mathcal{C}}$. Theorem 2.6.4 then follows.

In the following cases, the minimum distance of the BCH code $\mathcal{C}_{(q,n,\delta,b)}$ is known. We first have the following [1323, p. 260].

**Theorem 2.6.5** *For any $h$ with $1 \leq h \leq m-1$, the narrow-sense primitive BCH code $\mathcal{C}_{(q,q^m-1,q^h-1,1)}$ has minimum distance $d = q^h - 1$.*

It is easy to prove the following result [1247], which is a generalization of the classical result for the narrow-sense primitive case.

**Theorem 2.6.6** *The code $\mathcal{C}_{(q,n,\delta,b)}$ has minimum distance $d = \delta$ if $\delta$ divides* $\gcd(n, b-1)$.

The following is proved in [1245], which is a generalization of a classical result of Kasami and Lin for the case $q = 2$.

**Theorem 2.6.7** *Let $m \geq 3$ for $q = 2$, $m \geq 2$ for $q = 3$, and $m \geq 1$ for $q \geq 4$. For the narrow-sense primitive BCH code $\mathcal{C}_{(q,q^m-1,\delta,1)}$ with $\delta = q^m - q^{m-1} - q^i - 1$, where $(m-2)/2 \leq i \leq m - \lfloor m/3 \rfloor - 1$, the minimum distance $d = \delta$.*

Although it is notoriously difficult to find out the minimum distance of a BCH code in general, in a small number of cases other than the cases dealt with in the three theorems above, the minimum distance of the BCH code $\mathcal{C}_{(q,n,\delta,b)}$ is known. In several cases, the weight distribution of some BCH codes are known. Detailed information can be found in [551, 555, 1245, 1246, 1247, 1277].

### 2.6.2 The Dimensions of BCH Codes

The dimension of the BCH code $\mathcal{C}_{(q,n,\delta,b)}$ with defining set $T(b,\delta)$ in (2.2) is $n - |T(b,\delta)|$. Since $|T(b,\delta)|$ may have a very complicated relation with $n$, $q$, $b$ and $\delta$, the dimension of the BCH code cannot be given exactly in terms of these parameters. The best one can do in general is to develop tight lower bounds on the dimension of BCH codes. The next theorem introduces such bounds [1008, Theorem 5.1.7].

**Theorem 2.6.8** *Let $\mathcal{C}$ be an $[n, \kappa]$ BCH code over $\mathbb{F}_q$ of designed distance $\delta$. Then the following statements hold.*

(a) $\kappa \geq n - \mathrm{ord}_n(q)(\delta - 1)$.

(b) *If $q = 2$ and $\mathcal{C}$ is a narrow-sense BCH code, then $\delta$ can be assumed odd; furthermore if $\delta = 2w + 1$, then $\kappa \geq n - \mathrm{ord}_n(q)w$.*

The bounds in Theorem 2.6.8 may not be improved for the general case, as demonstrated by the following example. However, in some special cases, they could be improved.

**Example 2.6.9** Note that $m = \mathrm{ord}_{15}(2) = 4$, and the 2-cyclotomic cosets modulo 15 are

$$C_0 = \{0\},\ C_1 = \{1, 2, 4, 8\},\ C_3 = \{3, 6, 9, 12\},$$
$$C_5 = \{5, 10\},\ C_7 = \{7, 11, 13, 14\}.$$

Let $\gamma$ be a generator of $\mathbb{F}_{2^4}^*$ with $\gamma^4 + \gamma + 1 = 0$ and let $\alpha = \gamma^{(2^4-1)/15} = \gamma$ be the primitive $15^{\text{th}}$ root of unity.

When $(b, \delta) = (0, 3)$, the defining set $T(b, \delta) = \{0, 1, 2, 4, 8\}$, and the binary cyclic code has parameters $[15, 10, 4]$ and generator polynomial $x^5 + x^4 + x^2 + 1$. In this case, the actual minimum weight is more than the designed distance, and the dimension is larger than the bound in Theorem 2.6.8(a).

When $(b, \delta) = (1, 3)$, the defining set $T(b, \delta) = \{1, 2, 4, 8\}$, and the binary cyclic code has parameters $[15, 11, 3]$ and generator polynomial $x^4 + x + 1$. It is a narrow-sense BCH

code. In this case, the actual minimum weight is equal to the designed distance, and the dimension reaches the bound in Theorem 2.6.8(b).

When $(b, \delta) = (2, 3)$, the defining set $T(b, \delta) = \{1, 2, 3, 4, 6, 8, 9, 12\}$, and the binary cyclic code has parameters $[15, 7, 5]$ and generator polynomial $x^8 + x^7 + x^6 + x^4 + 1$. In this case, the actual minimum weight is more than the designed distance, and the dimension achieves the bound in Theorem 2.6.8(a).

When $(b, \delta) = (1, 5)$, the defining set $T(b, \delta) = \{1, 2, 3, 4, 6, 8, 9, 12\}$, and the binary cyclic code has parameters $[15, 7, 5]$ and generator polynomial $x^8 + x^7 + x^6 + x^4 + 1$. In this case, the actual minimum weight is equal to the designed distance, and the dimension is larger than the bound in Theorem 2.6.8(a). Note that the three pairs $(b_1, \delta_1) = (2, 3), (b_2, \delta_2) = (2, 4)$ and $(b_3, \delta_3) = (1, 5)$ define the same binary cyclic code with generator polynomial $x^8 + x^7 + x^6 + x^4 + 1$. Hence the maximum designed distance of this $[15, 7, 5]$ cyclic code is 5.

When $(b, \delta) = (3, 4)$, the defining set $T(b, \delta) = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 12\}$, and the binary cyclic code has parameters $[15, 5, 7]$ and generator polynomial $x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$. In this case, the actual minimum weight is more than the designed distance, and dimension is larger than the bound in Theorem 2.6.8(a).

The following is a general result on the dimension of BCH codes [47].

**Theorem 2.6.10** *Let* $\gcd(n, q) = 1$ *and* $q^{\lfloor m/2 \rfloor} < n \leq q^m - 1$, *where* $m = \mathrm{ord}_n(q)$. *Let* $2 \leq \delta \leq \min\{\lfloor nq^{\lceil m/2 \rceil}/(q^m - 1) \rfloor, n\}$. *Then*

$$\dim(\mathcal{C}_{(q,n,\delta,1)}) = n - m\lceil (\delta - 1)(1 - 1/q) \rceil.$$

Theorem 2.6.10 is useful when $n = q^m - 1$ or $n = (q^m - 1)/(q - 1)$, but may not be useful in some cases as the range for $\delta$ may be extremely small. It is in general very difficult to determine the dimensions of BCH codes. In a very small number of cases, the dimension of the BCH code $\mathcal{C}_{(q,n,\delta,1)}$ is known. For further information, the reader is referred to [551, 555, 1245, 1246, 1247, 1277].

### 2.6.3 Other Aspects of BCH Codes

The automorphism groups of BCH codes in most cases are open, but are known in some cases [161]. The weight distributions of the cosets of some BCH codes were considered in [386, 387, 388]. This problem is as hard as the determination of the weight distributions of BCH codes. The dual of a BCH code may not be a BCH code. An interesting problem is to characterise those BCH codes whose duals are also BCH codes.

Almost all references on BCH codes are about the primitive case. Only a few references on BCH codes with lengths $n = (q^m - 1)/(q - 1)$ or $n = q^\ell + 1$ exist in the literature [1246, 1247, 1277]. Most BCH codes have never been investigated. This is due to the fact that the $q$-cyclotomic cosets modulo $n$ are very irregular and behave very badly in most cases. For example, in most cases it is extremely difficult to determine the largest coset leader, not to mention the dimension and minimum distance of a BCH code. This partially explains the difficulty in researching into BCH codes. A characteristic of BCH codes is that it is hard in general to determine both the dimension and minimum distance of a BCH code.

## 2.7 Duadic Codes

Duadic codes are a family of cyclic codes and are generalizations of the quadratic residue codes. Binary duadic codes were defined in [1220] and were generalized to arbitrary finite fields in [1517, 1519]. Some duadic codes have very good parameters, while some have very bad parameters. The objective of this section is to give a brief introduction of duadic codes.

As before, let $n$ be a positive integer and $q$ a prime power with $\gcd(n, q) = 1$. Let $S_1$ and $S_2$ be two subsets of $\mathbb{Z}_n$ such that

- $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 = \mathbb{Z}_n \setminus \{0\}$, and
- both $S_1$ and $S_2$ are a union of some $q$-cyclotomic cosets modulo $n$.

If there is a unit $\mu \in \mathbb{Z}_n$ such that $S_1\mu = S_2$ and $S_2\mu = S_1$, then $(S_1, S_2, \mu)$ is called a **splitting** of $\mathbb{Z}_n$.

Recall that $m := \mathrm{ord}_n(q)$ and $\alpha$ is a primitive $n^{\text{th}}$ root of unity in $\mathbb{F}_{q^m}$. Let $(S_1, S_2, \mu)$ be a splitting of $\mathbb{Z}_n$. Define

$$g_i(x) = \prod_{i \in S_i} (x - \alpha^i) \quad \text{and} \quad \widetilde{g}_i(x) = (x - 1)g_i(x)$$

for $i \in \{1, 2\}$. Since both $S_1$ and $S_2$ are unions of $q$-cyclotomic cosets modulo $n$, both $g_1(x)$ and $g_2(x)$ are polynomials over $\mathbb{F}_q$. The pair of cyclic codes $\mathcal{C}_1$ and $\mathcal{C}_2$ of length $n$ over $\mathbb{F}_q$ with generator polynomials $g_1(x)$ and $g_2(x)$ are called **odd-like duadic codes**, and the pair of cyclic codes $\widetilde{\mathcal{C}}_1$ and $\widetilde{\mathcal{C}}_2$ of length $n$ over $\mathbb{F}_q$ with generator polynomials $\widetilde{g}_1(x)$ and $\widetilde{g}_2(x)$ are called **even-like duadic codes**.

By definition, $\mathcal{C}_1$ and $\mathcal{C}_2$ have parameters $[n, (n + 1)/2]$ and $\widetilde{\mathcal{C}}_1$ and $\widetilde{\mathcal{C}}_2$ have parameters $[n, (n-1)/2]$. For odd-like duadic codes, we have the following result [1008, Theorem 6.5.2].

**Theorem 2.7.1 (Square Root Bound)** *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be a pair of odd-like duadic codes of length $n$ over $\mathbb{F}_q$. Let $d_o$ be their (common) minimum odd-like weight. Then the following hold.*

(a) $d_o^2 \geq n$.

(b) *If the splitting defining the duadic codes is given by $\mu = -1$, then $d_o^2 - d_o + 1 \geq n$.*

(c) *Suppose $d_o^2 - d_o + 1 = n$, where $d_o > 2$, and assume that the splitting defining the duadic codes is given by $\mu = -1$. Then $d_o$ is the minimum weight of both $\mathcal{C}_1$ and $\mathcal{C}_2$.*

**Example 2.7.2** Let $(n, q) = (49, 2)$. Define

$$S_1 = \{1, 2, 4, 8, 9, 11, 15, 16, 18, 22, 23, 25, 29, 30, 32, 36, 37, 39, 43, 44, 46\} \cup \{7, 14, 28\}$$

and

$$S_2 = \{1, 2, \ldots, 48\} \setminus S_1.$$

It is easily seen that $(S_1, S_2, -1)$ is a splitting of $\mathbb{Z}_{49}$. The pair of odd-like duadic codes $\mathcal{C}_1$ and $\mathcal{C}_2$ defined by this splitting have parameters $[49, 25, 4]$ and generator polynomials

$$x^{24} + x^{22} + x^{21} + x^{10} + x^8 + x^7 + x^3 + x + 1$$

and

$$x^{24} + x^{23} + x^{21} + x^{17} + x^{16} + x^{14} + x^3 + x^2 + 1$$

respectively. The minimum weight of the two codes is even (i.e., 4), while the minimum odd-like weight in the two codes is 9. Note the lower bound on $d_o$ given in Theorem 2.7.1 is 7.

Duadic codes of prime lengths are of special interest as they include the quadratic residue codes, which are defined as follows.

**Definition 2.7.3** Let $n$ be an odd prime and $q$ be a quadratic residue modulo $n$. Denote by $S_1$ and $S_2$ the set of quadratic residues and the set of quadratic non-residues, respectively. Let $\mu$ be an element of $S_2$. Then $(S_1, S_2, \mu)$ is a splitting of $\mathbb{Z}_n$. The corresponding four duadic codes $\mathcal{C}_1, \mathcal{C}_2, \widetilde{\mathcal{C}_1}, \widetilde{\mathcal{C}_2}$ are called **quadratic residue codes.**

It is known that the automorphism groups of the extended odd-like quadratic residue codes are transitive. Hence, their minimum weight codewords must be odd-like. We then have the following.

**Theorem 2.7.4 (Square Root Bound)** *Let $n$ be an odd prime and $q$ be a quadratic residue modulo $n$. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be a pair of odd-like quadratic residue codes of length $n$ over $\mathbb{F}_q$. Let $d$ be their (common) minimum weight. Then the following hold.*

(a) $d^2 \geq n$.

(b) *If $-1$ is a quadratic non-residue, then $d^2 - d + 1 \geq n$.*

(c) *Suppose $d^2 - d + 1 = n$, where $d > 2$, and assume that $-1$ is a quadratic non-residue. Then $d$ is the minimum weight of both $\mathcal{C}_1$ and $\mathcal{C}_2$.*

The Golay codes $\mathcal{C}_1$ and $\mathcal{C}_2$ introduced in Section 1.13 of Chapter 1 are the odd-like quadratic residue binary codes of length 23 and have parameters $[23, 12, 7]_2$. The corresponding even-like quadratic residue codes have parameters $[23, 11, 8]_2$. The ternary Golay codes described in Section 1.13 are also quadratic residue codes.

It is very hard to determine the minimum distance of quadratic residue codes. However, the Square Root Bound on their minimum distances is good enough. Quadratic residue codes are interesting partly because their extended codes hold 2-designs and 3-designs. Quadratic residue codes of length the product of two primes were introduced in [546].

Under certain conditions the extended odd-like duadic codes are self-dual [1220]. Duadic codes have a number of interesting properties. For further information on the existence, constructions, and properties of duadic codes, the reader is referred to [1008, Chapter 6], [546], [559], and [565].

## 2.8 Punctured Generalized Reed–Muller Codes

Binary Reed–Muller codes were introduced in Section 1.11. It is known that these codes are equivalent to the extended codes of some cyclic codes. In other words, after puncturing the binary Reed–Muller codes at a proper coordinate, the obtained codes are permutation equivalent to some cyclic codes. The purpose of this section is to introduce a family of cyclic codes of length $n = q^m - 1$ over $\mathbb{F}_q$ whose extended codes are the generalized Reed–Muller code over $\mathbb{F}_q$.

Let $q$ be a prime power as before. For any integer $j = \sum_{i=0}^{m-1} j_i q^i$, where $0 \le j_i \le q-1$ for all $0 \le i \le m-1$ and $m$ is a positive integer, we define

$$\omega_q(j) = \sum_{i=0}^{m-1} j_i,$$

where the sum is taken over the ring of integers, and is called the **$q$-weight** of $j$.

Let $\ell$ be a positive integer with $1 \le \ell < (q-1)m$. The $\ell^{\text{th}}$ order **punctured generalized Reed–Muller code** $\mathcal{RM}_q(\ell, m)^*$ over $\mathbb{F}_q$ is the cyclic code of length $n = q^m - 1$ with generator polynomial

$$g(x) = \prod_{\substack{1 \le j \le n-1 \\ \omega_q(j) < (q-1)m-\ell}} (x - \alpha^j),$$

where $\alpha$ is a generator of $\mathbb{F}_{q^m}^*$. Since $\omega_q(j)$ is a constant function on each $q$-cyclotomic coset modulo $n = q^m - 1$, $g(x)$ is a polynomial over $\mathbb{F}_q$.

The parameters of the punctured generalized Reed–Muller code $\mathcal{RM}_q(\ell, m)^*$ are known and summarized in the next theorem [71, Section 5.5].

**Theorem 2.8.1** *For any $\ell$ with $0 \le \ell < (q-1)m$, $\mathcal{RM}_q(\ell, m)^*$ is a cyclic code over $\mathbb{F}_q$ with length $n = q^m - 1$, dimension*

$$\kappa = \sum_{i=0}^{\ell} \sum_{j=0}^{m} (-1)^j \binom{m}{j} \binom{i - jq + m - 1}{i - jq}$$

*and minimum weight $d = (q - \ell_0)q^{m-\ell_1-1} - 1$, where $\ell = \ell_1(q-1) + \ell_0$ and $0 \le \ell_0 < q-1$.*

**Example 2.8.2** Let $(q, m, \ell) = (3, 3, 3)$, and let $\alpha$ be a generator of $\mathbb{F}_{3^3}^*$ with $\alpha^3 + 2\alpha + 1 = 0$. Then $\mathcal{RM}_3(3, 3)^*$ is a ternary code with parameters $[26, 17, 5]$ and generator polynomial

$$g(x) = x^9 + 2x^8 + x^7 + x^6 + x^5 + 2x^4 + 2x^3 + 2x^2 + x + 1.$$

The dual of the punctured generalized Reed–Muller code is described in the following theorem [71, Corollary 5.5.2].

**Theorem 2.8.3** *For $0 \le \ell < m(q-1)$, the code $(\mathcal{RM}_q(\ell, m)^*)^\perp$ is the cyclic code with generator polynomial*

$$h(x) = \prod_{\substack{0 \le j \le n-1 \\ \omega_q(j) \le \ell}} (x - \alpha^j),$$

*where $\alpha$ is a generator of $\mathbb{F}_{q^m}^*$. In addition,*

$$(\mathcal{RM}_q(\ell, m)^*)^\perp = (\mathbb{F}_q \mathbf{1})^\perp \cap \mathcal{RM}_q(m(q-1) - 1 - \ell, m)^*,$$

*where $\mathbf{1}$ is the all-one vector in $\mathbb{F}_q^n$ and $\mathbb{F}_q\mathbf{1}$ denotes the code over $\mathbb{F}_q$ with length $n$ generated by $\mathbf{1}$.*

**Example 2.8.4** Let $(q, m, \ell) = (3, 3, 3)$, and let $\alpha$ be a generator of $\mathbb{F}_{3^3}^*$ with $\alpha^3 + 2\alpha + 1 = 0$. Then $(\mathcal{RM}_3(3, 3)^*)^\perp$ is a ternary code with parameters $[26, 9, 9]$ and generator polynomial

$$\begin{aligned} g^\perp(x) \;=\; & x^{17} + 2x^{16} + 2x^{15} + x^{14} + x^{13} + x^{11} + 2x^{10} + 2x^9 + \\ & x^8 + 2x^7 + 2x^5 + x^4 + 2x^3 + 2x + 2. \end{aligned}$$

The codes $\mathcal{RM}_q(\ell, m)^*$ have a geometric interpretation. Their extended codes hold 2-designs for $q > 2$, and 3-designs for $q = 2$. The reader is referred to [71, Corollary 5.2] for further information.

## 2.9 Another Generalization of the Punctured Binary Reed–Muller Codes

The punctured generalized Reed–Muller codes are a generalization of the classical punctured binary Reed–Muller codes, and were introduced in the previous section. A new generalization of the classical punctured binary Reed–Muller codes was given recently in [561]. The task of this section is to introduce the newly generalized cyclic codes.

Let $n = q^m - 1$. For any integer $a$ with $0 \le a \le n-1$, we have the following $q$-adic expansion

$$a = \sum_{j=0}^{m-1} a_j q^j,$$

where $0 \le a_j \le q-1$. The Hamming weight of $a$, denoted by $\mathrm{wt}_{\mathrm{H}}(a)$, is the number of nonzero coordinates in the vector $(a_0, a_1, \ldots, a_{m-1})$.

Let $\alpha$ be a generator of $\mathbb{F}_{q^m}^*$. For any $1 \le h \le m$, we define a polynomial

$$g_{(q,m,h)}(x) = \prod_{\substack{1 \le a \le n-1 \\ 1 \le \mathrm{wt}_{\mathrm{H}}(a) \le h}} (x - \alpha^a).$$

Since $\mathrm{wt}_{\mathrm{H}}(a)$ is a constant function on each $q$-cyclotomic coset modulo $n$, $g_{(q,m,h)}(x)$ is a polynomial over $\mathbb{F}_q$. By definition, $g_{(q,m,h)}(x)$ is a divisor of $x^n - 1$.

Let $\mathcal{U}(q, m, h)$ denote the cyclic code over $\mathbb{F}_q$ with length $n$ and generator polynomial $g_{(m,q,h)}(x)$. By definition, $g_{(q,m,m)}(x) = (x^n - 1)/(x - 1)$. Therefore, the code $\mathcal{U}(q, m, m)$ is trivial, as it has parameters $[n, 1, n]$ and is spanned by the all-1 vector. Below we consider the code $\mathcal{U}(q, m, h)$ for $1 \le h \le m-1$ only.

**Theorem 2.9.1** *Let $m \ge 2$ and $1 \le h \le m-1$. Then $\mathcal{U}(q, m, h)$ has parameters $[q^m - 1, \kappa, d]$, where*

$$\kappa = q^m - \sum_{i=0}^{h} \binom{m}{i} (q-1)^i$$

*and*

$$\frac{q^{h+1} - 1}{q - 1} \le d \le 2q^h - 1. \tag{2.3}$$

When $q = 2$, the code $\mathcal{U}(q, m, h)$ clearly becomes the classical punctured binary Reed–Muller code $\mathcal{RM}(m-1-h, m)^*$. Hence, $\mathcal{U}(q, m, h)$ is indeed a generalization of the original punctured binary Reed–Muller code. In addition, when $q = 2$, the lower bound and the upper bound in (2.3) become identical. It is conjectured that the lower bound on $d$ is the actual minimum distance.

**Example 2.9.2** The following is a list of examples of the code $\mathcal{U}(q, m, h)$.

1. When $(q, m, h) = (3, 3, 1)$, $\mathcal{U}(q, m, h)$ has parameters $[26, 20, 4]$, and is distance-optimal.
2. When $(q, m, h) = (3, 4, 1)$, $\mathcal{U}(q, m, h)$ has parameters $[80, 72, 4]$, and is distance-optimal.
3. When $(q, m, h) = (3, 4, 2)$, $\mathcal{U}(q, m, h)$ has parameters $[80, 48, 13]$, and its minimum distance is one less than that of the best code with parameters $[80, 48, 14]$.

4. When $(q, m, h) = (3, 4, 3)$, the code $\mathcal{U}(q, m, h)$ has parameters $[80, 16, 40]$, which are the best parameters known.

5. When $(q, m, h) = (4, 3, 1)$, the code $\mathcal{U}(q, m, h)$ has parameters $[63, 54, 5]$, which are the best parameters known.

An interesting fact about the family of newly generalized codes $\mathcal{U}(q, m, h)$ is the following.

**Corollary 2.9.3** *Let $m \geq 2$. Then the ternary code $\mathcal{U}(3, m, 1)$ has parameters $[3^m - 1, 3^m - 1 - 2m, 4]$ and is distance-optimal.*

We have also the next two special cases in which the parameters of the code $\mathcal{U}(q, m, h)$ are known.

**Theorem 2.9.4** *Let $m$ be even. Then the cyclic code $\mathcal{U}(q, m, 1)$ has parameters $[q^m - 1, q^m - 1 - m(q-1), q+1]$.*

**Theorem 2.9.5** *Let $m \geq 2$. Then the cyclic code $\mathcal{U}(q, m, m-1)$ has parameters $[q^m - 1, (q-1)^m, (q^m - 1)/(q-1)]$.*

The following theorem gives information on the parameters of the dual code $\mathcal{U}(q, m, h)^{\perp}$.

**Theorem 2.9.6** *Let $m \geq 2$ and $1 \leq h \leq m-1$. The dual code $\mathcal{U}(q, m, h)^{\perp}$ has parameters $[q^m - 1, \kappa^{\perp}, d^{\perp}]$, where*

$$\kappa^{\perp} = \sum_{i=1}^{h} \binom{m}{i} (q-1)^i.$$

*The minimum distance $d^{\perp}$ of $\mathcal{U}(q, m, h)^{\perp}$ is bounded below by*

$$d^{\perp} \geq q^{m-h} + q - 2.$$

When $q = 2$, the lower bound on the minimum distance $d^{\perp}$ of $\mathcal{U}(q, m, h)^{\perp}$ given in Theorem 2.9.6 is achieved. Experimental data shows that the lower bound on $d^{\perp}$ in Theorem 2.9.6 is not tight for $q > 2$. It is open how to improve it or determine the exact minimum distance.

The code $\mathcal{U}(q, m, h)$ is clearly different from the punctured generalized Reed–Muller code. It is open if $\mathcal{U}(q, m, h)$ has a geometric interpretation. For a proof of the results introduced above and further properties of the cyclic code $\mathcal{U}(q, m, h)$, the reader is referred to [561].

## 2.10 Reversible Cyclic Codes

**Definition 2.10.1** A linear code $\mathcal{C}$ is **reversible**[1] if $(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$ implies that $(c_{n-1}, c_{n-2}, \ldots, c_0) \in \mathcal{C}$.

[1] A linear code of length $n$ over $\mathbb{F}_q$ is called an **LCD code (linear code with complementary dual)** if $\mathcal{C} \cap \mathcal{C}^{\perp} = \{\mathbf{0}\}$, which is equivalent to $\mathcal{C} \oplus \mathcal{C}^{\perp} = \mathbb{F}_q^n$. Reversible cyclic codes are in fact LCD codes.

Reversible cyclic codes were considered in [1346, 1347]. A cryptographic application of reversible cyclic codes was proposed in [353]. A well rounded treatment of reversible cyclic codes was given in [1236]. The objective of this section is to deliver a basic introduction to reversible cyclic codes.

**Definition 2.10.2** A polynomial $f(x)$ over $\mathbb{F}_q$ is called **self-reciprocal** if it equals its reciprocal $f^{\perp}(x)$.

The conclusions of the following theorem are known in the literature [1323, page 206] and are easy to prove.

**Theorem 2.10.3** *Let $\mathcal{C}$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with generator polynomial $g(x)$. Then the following statements are equivalent.*

(a) *$\mathcal{C}$ is reversible.*

(b) *$g(x)$ is self-reciprocal.*

(c) *$\beta^{-1}$ is a root of $g(x)$ for every root $\beta$ of $g(x)$ over the splitting field of $g(x)$.*

*Furthermore, if $-1$ is a power of $q$ mod $n$, then every cyclic code over $\mathbb{F}_q$ of length $n$ is reversible.*

Now we give an exact count of reversible cyclic codes of length $n = q^m - 1$ for odd primes $m$. Recall the $q$-cyclotomic cosets $C_a$ modulo $n$ given in Definition 1.12.7. It is straightforward that $-a = n - a \in C_a$ if and only if $a(1 + q^j) \equiv 0 \pmod{n}$ for some integer $j$. The following two lemmas are straightforward and hold whenever $\gcd(n, q) = 1$.

**Lemma 2.10.4** *The irreducible polynomial $M_{\alpha^a}(x)$ is self-reciprocal if and only if $n - a \in C_a$.*

**Lemma 2.10.5** *The least common multiple $\operatorname{lcm}(M_{\alpha^a}(x), M_{\alpha^{n-a}}(x))$ is self-reciprocal for every $a \in \mathbb{Z}_n$.*

**Definition 2.10.6** The least nonnegative integer in a $q$-cyclotomic coset modulo $n$ is called the **coset leader** of this coset.

By Lemma 2.10.4, we have that

$$\operatorname{lcm}(M_{\alpha^a}(x), M_{\alpha^{n-a}}(x)) = \begin{cases} M_{\alpha^a}(x) & \text{if } n - a \in C_a, \\ M_{\alpha^a}(x) M_{\alpha^{n-a}}(x) & \text{otherwise.} \end{cases}$$

Let $\Gamma_{(n,q)}$ denote the set of coset leaders of all $q$-cyclotomic cosets modulo $n$. Define

$$\Pi_{(n,q)} = \Gamma_{(n,q)} \setminus \left\{ \max\{a, \text{leader}(n-a)\} \mid a \in \Gamma_{(n,q)}, n - a \notin C_a \right\},$$

where leader$(i)$ denotes the coset leader of $C_i$. Then $\{C_a \cup C_{n-a} \mid a \in \Pi_{(q,n)}\}$ is a partition of $\mathbb{Z}_n$.

The following conclusion then follows directly from Lemmas 2.10.4, 2.10.5, and Theorem 2.10.3.

**Theorem 2.10.7** *The total number of reversible cyclic codes over $\mathbb{F}_q$ of length $n$ is equal to $2^{|\Pi_{(q,n)}|}$, including the zero code and the code $\mathbb{F}_q^n$. Every reversible cyclic code over $\mathbb{F}_q$ of length $n$ is generated by a polynomial*

$$g(x) = \prod_{a \in S} \operatorname{lcm}\left(M_{\alpha^a}(x), M_{\alpha^{n-a}}(x)\right),$$

*where $S$ is a (possibly empty) subset of $\Pi_{(q,n)}$.*

**Example 2.10.8** Let $(n, q) = (15, 2)$. The 2-cyclotomic cosets modulo 15 are

$$C_0 = \{0\},\ C_1 = \{1, 2, 4, 8\},\ C_3 = \{3, 6, 9, 12\},\ C_5 = \{5, 10\},\ \text{and } C_7 = \{7, 11, 13, 14\}.$$

We also have

$$x^{15} - 1 = M_{\alpha^0}(x)M_{\alpha^1}(x)M_{\alpha^3}(x)M_{\alpha^5}(x)M_{\alpha^7}(x),$$

where

$$\begin{aligned} M_{\alpha^0}(x) &= x + 1, \\ M_{\alpha^1}(x) &= x^4 + x + 1, \\ M_{\alpha^3}(x) &= x^4 + x^3 + x^2 + x + 1, \\ M_{\alpha^5}(x) &= x^2 + x + 1, \\ M_{\alpha^7}(x) &= x^4 + x^3 + 1. \end{aligned}$$

Note that $M_{\alpha^i}(x)$ are self-reciprocal for $i \in \{0, 3, 5\}$ while $M_{\alpha^1}(x)$ and $M_{\alpha^7}(x)$ are reciprocals of each other. In this case,

$$\Gamma_{(n,q)} = \{0, 1, 3, 5, 7\}.$$

But

$$\Pi_{(n,q)} = \{0, 1, 3, 5\}.$$

Hence, there are 16 reversible binary cyclic codes of length 15, including the zero code and the code $\mathbb{F}_2^{15}$.

**Corollary 2.10.9** *Let $q$ be an even prime power and $n = q^m - 1$. If $m$ is odd, then the only self-reciprocal irreducible divisor of $x^n - 1$ over $\mathbb{F}_q$ is $x - 1$. If $m$ is an odd prime, then the total number of reversible cyclic codes of length $n$ over $\mathbb{F}_q$ is equal to $2^{\frac{q^m+(m-1)q}{2m}}$, including the zero code and the code $\mathbb{F}_q^n$.*

**Corollary 2.10.10** *Let $q$ be an odd prime power and $n = q^m - 1$. If $m$ is odd, then the only self-reciprocal irreducible divisors of $x^n - 1$ over $\mathbb{F}_q$ are $x - 1$ and $x + 1$. If $m$ is an odd prime, then the total number of reversible cyclic codes of length $n$ over $\mathbb{F}_q$ is equal to $2^{\frac{q^m+(m-1)q+m}{2m}}$, including the zero code and the code $\mathbb{F}_q^n$.*

The following three theorems follow directly from Theorem 2.10.3 and the definition of BCH codes, and can be viewed as corollaries of Theorem 2.10.3.

**Theorem 2.10.11** *The BCH code $\mathcal{C}_{(q,n,\delta,b)}$ is reversible if $b = -t$ and the designed distance is $\delta = 2t + 2$ for any nonnegative integer $t$.*

**Theorem 2.10.12** *The BCH code $\mathcal{C}_{(q,n,\delta,b)}$ is reversible if $n$ is odd, $b = (n - t)/2$ and the designed distance is $\delta = t + 2$ for any odd integer $t$ with $1 \le t \le n - 2$.*

**Theorem 2.10.13** *The BCH code $\mathcal{C}_{(q,n,\delta,b)}$ is reversible if $n$ is even, $b = (n - 2t)/2$ and the designed distance is $\delta = 2t + 2$ for any integer $t$ with $0 \le t \le n/2$.*

The dimensions of some of the reversible BCH codes described in Theorems 2.10.11, 2.10.12, and 2.10.13 were determined in [1236]. Two families of reversible BCH codes were studied in [1247]. Non-primitive reversible cyclic codes were also treated in [1236]. There are many reversible cyclic codes and it is easy to construct them. However, determining their parameters is a difficult problem in general. There are clearly reversible cyclic codes with both good and bad parameters.

# Chapter 3

# Construction and Classification of Codes

**Patric R. J. Östergård**
*Aalto University*



## 3.1 Introduction

Shannon's seminal work [1661] showed in a nonconstructive way that good codes exist, as discussed in Section 1.1, and marked the birth of coding and information theory. One of the main goals of coding theory is to construct codes that are as good as possible for given parameters and types of codes. There is a mathematical as well as an engineering dimension of the construction problem. Whereas mathematicians may wish to study arbitrary parameters and the behavior of codes when the length tends to infinity, the parameters of a code used in some engineering applications are necessarily fixed and bounded. Depending on one's philosophical viewpoint, one may say that codes are discovered rather than constructed. Hence, perhaps the most fundamental question in coding theory can also be phrased in terms of **existence**: Does a code with given parameters exist or not?

If the answer to an existence question is affirmative, one may further wish to carry out a **classification** of those codes, that is, to determine all possible such codes up to symmetry (equivalence, isomorphism). Also classification has both a theoretical and a practical side. Classified codes can be studied to gain more insight into codes, to state or refute conjectures, and so on. But they also form an exhaustive set of candidate codes, whose performance can be tested in actual applications.

**Remark 3.1.1** A nonexistence result is intrinsically a classification result, where the outcome is the empty set.

A lot of (manual and/or computational) resources may be required for the study of existence or classification of codes with certain parameters. However, whereas verifying correctness of a (positive) outcome is fast and straightforward for a constructed code, it can be as time-consuming as the original work for a set of classified codes.

Validation of classification results is considered in [1090, Chap. 10]. Especially double-counting techniques have turned out to be useful. If we know the total number of codes—for example, through a *mass formula*—then we can check whether the orbit-stabilizer theorem applied to a set of classified codes leads to the same number; see for example Theorem 4.5.2. This approach can even be used to show that a set of codes found in a non-exhaustive search is complete. Lam and Thiel [1192] showed how double-counting can be integrated into classification algorithms when no mass formula is available (which is the typical situation).

The main classical reference in coding theory, *The Theory of Error-Correcting Codes* by MacWilliams and Sloane [1323], considers the existence problem extensively and is still, many decades after the publication of the first edition, a good source of information. The theme of classifying codes, on the other hand, has flourished much later, in part because of its high dependence on computational resources; the monograph [1090] provides an in-depth treatment of this theme.

In this chapter, a general overview of existence and classification problems will be given, and some highlights from the past will be singled out. The chapter touches upon several of the problems and problem areas presented in the list of major open problems in algebraic coding theory in [630].

## 3.2 Equivalence and Isomorphism

The concepts of equivalence and isomorphism of codes are briefly discussed in Section 1.8. Generally, the term *symmetry* covers both of those concepts, especially when considering maps from a code onto itself, that is, automorphisms. Namely, such maps lead to groups under composition, and groups are essentially about symmetries. The group formed by all automorphisms of a code is, whenever the type of automorphisms is understood, simply called the **automorphism group** of the code. A subgroup of the automorphism group is called a **group of automorphisms**.

Symmetries play a central role when constructing as well as classifying codes: several types of constructions are essentially about prescribing symmetries, and one core part of classification is about dealing with maps and symmetries.

On a high level of abstraction, the same questions are asked for linear and unrestricted codes and analogous techniques are used. On a detailed level, however, there are significant differences between those two types of codes.

Consider codes of length $n$ over $\mathbb{F}_q$. We have seen in Definition 1.8.8 that equivalence of unrestricted codes is about permuting coordinates and the elements of the alphabet, individually within each coordinate. All such maps form a group that is isomorphic to the wreath product $\mathrm{S}_q \wr \mathrm{S}_n$. For linear codes on the other hand, the concepts of permutation equivalence, monomial equivalence, and equivalence lead to maps that form groups isomorphic to $\mathrm{S}_n$, $\mathbb{F}_q^* \wr \mathrm{S}_n$, and the semidirect product $(\mathbb{F}_q^* \wr \mathrm{S}_n) \rtimes_\theta \mathrm{Aut}(\mathbb{F}_q)$, respectively, where $\mathbb{F}_q^*$ is the multiplicative group of $\mathbb{F}_q$ and $\theta : \mathrm{Aut}(\mathbb{F}_q) \to \mathrm{Aut}(\mathbb{F}_q^* \wr \mathrm{S}_n)$ is a group homomorphism.

**Remark 3.2.1** For *binary* linear codes, all three types of equivalence coincide.

### 3.2.1 Prescribing Symmetries

A code of size $M$ is a subset of $M$ vectors from the $n$-dimensional vector space over $\mathbb{F}_q$ which fulfills some requirements depending on the type of code. The number of ways to choose $M$ arbitrary vectors from such a space is $\binom{q^n}{M}$, which becomes astronomically large already for rather small parameters. (This is obviously the total number of $(n, M)_q$ codes.) Although no general conclusion regarding the hardness of solving construction and classification problems can be drawn from this number, the number does give a clue that the limit of what is feasible might be reached quite early. Indeed, this is what happens, but perhaps not as early as one would think.

**Example 3.2.2** In some special cases—in particular, for perfect codes—quite large unrestricted codes have been classified, such as the $(23, 4096, 7)_2$ code (the binary Golay code is unique [1732]; see also [525]) and the $(15, 2048, 3)_2$ codes (with the parameters of a Hamming code; there are 5983 such codes [1472]).

But what can be done if we go beyond parameters for which the size of an optimal code can be determined and the optimal codes can be classified? Analytical upper bounds and constructive lower bounds on the size of codes can still be used. One way to speed up computer-aided constructive techniques—some of which are discussed in Chapter 23—is to restrict the search by imposing a structure on the codes. This is a two-edged sword: the search space is reduced, but good codes might not have that particular structure. Hence some experience is of great help in tuning the search. A very common approach is that of prescribing symmetries (automorphisms).

**Remark 3.2.3** In the discussion of groups in the context of automorphism groups of codes, we are not only interested in the abstract group but in the group and its action. This is implicitly understood in the sequel when talking about one particular group or all groups of certain orders. For example, "prescribing a group" means "prescribing a group and its action" and "considering all groups" means "considering all groups and all possible actions of those groups".

By prescribing a group $G$, the $n$-dimensional vector space is partitioned into orbits of vectors. The construction problem then becomes a problem of finding a set of those orbits rather than finding a set of individual vectors. It must further be checked that the orbits themselves are feasible; an orbit whose codewords do not fulfill the minimum distance criterion can be discarded immediately.

**Remark 3.2.4** An $[n, k]_q$ linear code can be viewed as an unrestricted code which contains the all-zero codeword and has a particular group of automorphisms $G$ of order $q^k$, which only permutes elements of the alphabet, individually within each coordinate.

**Example 3.2.5** Consider the $[4, 2]_2$ binary linear code generated by

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

If this code contains some word $\mathbf{c}$, then it also contains, for example, $\mathbf{c}+1010$. Adding 0 to a coordinate value means applying a value permutation that is the identity permutation, and adding 1 to a coordinate value means applying the value permutation $0 \leftrightarrow 1$. Permuting elements of the alphabet, individually within each coordinate, is indeed covered by the definition of equivalence of unrestricted codes.

Unrestricted codes that consist of cosets of a linear code can in a similar manner be considered in the framework of prescribed groups of automorphisms. Let $H$ be an $(n-k)\times n$ parity check matrix for an $[n,k,d']_q$ linear code[1], let $\mathbf{x},\mathbf{y}\in\mathbb{F}_q^{n-k}$, and let $\mathcal{S}\subseteq\mathbb{F}_q^{n-k}$. Then, if $\mathrm{wt_H}(\mathbf{t})$ is the Hamming weight of $\mathbf{t}$, we define

$$\mathrm{d}^H(\mathbf{x},\mathbf{y}) = \min\{\mathrm{wt_H}(\mathbf{t}) \mid H\mathbf{t}^\mathsf{T} = (\mathbf{x}-\mathbf{y})^\mathsf{T},\ \mathbf{t}\in\mathbb{F}_q^n\},$$
$$d^H(\mathcal{S}) = \min_{\mathbf{a},\mathbf{b}\in\mathcal{S},\mathbf{a}\neq\mathbf{b}} \mathrm{d}^H(\mathbf{a},\mathbf{b}).$$

The following theorem [1470, Theorem 1] shows that the situation here is a generalization of the standard approach of finding arbitrary codes with minimum distance $2r+1$ by packing Hamming balls of radius $r$. Here we are packing more general geometrical objects given by the columns of $H$, and we get the standard approach when $k=0$ and thereby $H=I_n$.

**Theorem 3.2.6** *Let $H$ be an $(n-k)\times n$ parity check matrix for an $[n,k,d']_q$ linear code, and let $\mathcal{S}\subseteq\mathbb{F}_q^{n-k}$. Then the code $\mathcal{C}=\{\mathbf{c}\in\mathbb{F}_q^n \mid H\mathbf{c}^T\in\mathcal{S}\}$ has minimum distance $\min\{d^H(\mathcal{S}),d'\}$.*

When searching for unrestricted codes, one may consider any subgroup of $\mathrm{S}_q\wr\mathrm{S}_n$. Due to the large number of (conjugacy classes of) subgroups, it is typically necessary to restrict the set of subgroups considered. The subgroups should not be too small, whence the search would not be limited enough, and not too big either. Experiments and experience, especially regarding automorphism groups of known good codes, are helpful in the process of sifting candidate groups. For unrestricted codes, Theorem 3.2.6 has turned out to be very useful. Groups that act transitively on the $n$ coordinates or even on the $qn$ coordinate–value pairs have also been considered with success [1186].

Arguably the principal automorphism for good linear as well as unrestricted codes is a cyclic permutation of the coordinates. Codes with such an automorphism are called cyclic. Cyclic *linear* codes—which have a nice algebraic structure as they can be viewed as ideals of certain quotient rings—are discussed in Section 1.12 and Chapter 2.

We have seen a generalization of cyclic codes to $l$-quasi-cyclic codes in Definition 1.12.23, and these are covered in Chapter 7; cyclic codes are 1-quasi-cyclic. Other possible automorphisms that are generalizations of the cyclic ones and that are common amongst good codes are given by the following definition. The definition applies to both linear and unrestricted codes. For linear codes, these types of codes can be considered algebraically. See also Chapter 17.

**Definition 3.2.7** Fix $\alpha\in\mathbb{F}_q^*$. A code $\mathcal{C}$ is called **$\alpha$-constacyclic** (or **$\alpha$-twisted**) if $c_0c_1\cdots c_{n-1}\in\mathcal{C}$ implies that $(\alpha c_{n-1})c_0c_1\cdots c_{n-2}\in\mathcal{C}$. A 1-constacyclic code is cyclic. A code $\mathcal{C}$ is **$(\alpha,l)$-quasi-twisted** if $l$ divides $n$ and $c_0c_1\cdots c_{n-1}\in\mathcal{C}$ implies that $(\alpha c_{n-l})(\alpha c_{n-l+1})\cdots(\alpha c_{n-1})c_0c_1\cdots c_{n-l-1}\in\mathcal{C}$. An $(\alpha,1)$-quasi-twisted code is $\alpha$-constacyclic (or $\alpha$-twisted).

Important types of codes with a large automorphism group include quadratic residue codes.

**Definition 3.2.8** Let $p$ be an odd prime, let $Q$ be the set of quadratic residues modulo $p$, let $q\in Q$ be a prime, and let $\zeta$ be a primitive $p^{\text{th}}$ root of unity in some finite extension field of $\mathbb{F}_q$. A **quadratic residue code of length $p$ over $\mathbb{F}_q$** is a cyclic code with generator polynomial

$$f(x)=\prod_{j\in Q}(x-\zeta^j).$$

[1]This means that $H$ has full rank, which is a reasonable assumption. We get essentially the same main theorem when $H$ does not have full rank, but then some details regarding distances have to be tuned.

**Remark 3.2.9** The quadratic residue codes in Definition 3.2.8 can be generalized to codes whose alphabet size is a prime power and to codes whose length is a prime power.

The dimension of a quadratic residue code of length $p$ is $(p+1)/2$. The minimum distance is known to be at least $\sqrt{p}$, which can be further strengthened for various parameters; see Theorem 2.7.4. The true minimum distance typically has to be determined on a case-by-case basis.

**Example 3.2.10** The $[7, 4, 3]_2$ binary Hamming code, the $[23, 12, 7]_2$ binary Golay code, and the $[11, 6, 5]_3$ ternary Golay code are quadratic residue codes.

Gleason and Prange showed that the automorphism group of an extended quadratic residue code is rather large. The result was originally published only in a laboratory research report, but it has later been discussed in most coding theory textbooks, including [1323, Chap. 16]. See also [210, 1002]. When extending quadratic residue codes, the standard Definition 1.7.1 can be used for $q = 2, 3$, but a different definition has to be used for general values of $q$; see, for example, [1323, Chap. 16] for details.

**Theorem 3.2.11 (Gleason–Prange)** *Every extended quadratic residue code of length $p$ has a group of automorphisms that is isomorphic to the projective special linear group* $\mathrm{PSL}_2(p)$.

**Remark 3.2.12** The extensions of the three codes in Example 3.2.10 are exceptional examples of extended quadratic residue codes whose automorphism group has $\mathrm{PSL}_2(p)$ as a proper subgroup.

There is a two-way interaction between codes and groups: known good codes (or families of good codes) can be studied to find their automorphism groups, and groups can be prescribed to find good codes with such automorphisms.

A search for codes with prescribed automorphisms that has a negative outcome leads to results of scientific value only if the search is exhaustive and the parameters are of general interest. For the most important open existence problems, smaller and smaller orders of automorphisms and groups of automorphisms are commonly considered in a sequence of publications. The hope in each and every such study is clearly to find codes with the prescribed groups of automorphisms.

**Remark 3.2.13** If the answer to a general existence problem is negative, then even a proof that a code cannot have nontrivial automorphisms does not essentially bring us closer to a nonexistence proof. However, this is not work in vain, but forms a strong basis for making a nonexistence conjecture.

**Example 3.2.14** Neil Sloane [1726] asked in 1973 whether there is a self-dual doubly-even $[72, 36, 16]_2$ code. This would be the third code in a sequence of self-dual doubly-even $[24m, 12m, 4m + 4]_2$ codes, which begins with the extended binary Golay code ($m = 1$) and the extended binary quadratic residue code of length 48 ($m = 2$). The latter code is a unique such code [991]; the extended binary Golay code is unique as a general linear code [1514] and even as an unrestricted code [1732].

Jessie MacWilliams, in her review of [1726] for *Mathematical Reviews*, mentions that the author had offered \$10 for a solution and added another \$10 to the prize money. It was thus clear from the very beginning that this is an important and interesting problem. Almost half a century later, the problem is still open, no less fascinating, and arguably the most important open specific case in the theory of linear codes.

Conway and Pless [437] showed that the possible prime orders for automorphisms of a self-dual doubly-even $[72, 36, 16]_2$ code are 2, 3, 5, 7, 11, 17, and 23. Starting from the biggest order, the largest four orders have been eliminated in [1516], [1525], [1010], and [723], respectively. After further elimination of various groups of orders $2^i 3^j 5^k$, $i, j, k \geq 0$ in [238, 239, 241, 274, 280, 1422, 1922, 1930, 1931], the groups of order at most 5, except for the cyclic group of order 4, remain. Moreover, automorphisms of order 2 and 3 cannot have fixed points [273, 274], and automorphisms of order 5 must have exactly two fixed points [437]. See also [240] and Section 4.3.

### 3.2.2 Determining Symmetries

The obvious recurrent specific questions when studying equivalence (or isomorphism) of (linear and unrestricted) codes and the symmetries of such codes are the following:

1. Given two codes, $\mathcal{C}_1$ and $\mathcal{C}_2$, are these equivalent (isomorphic) or not?
2. Given a code $\mathcal{C}$, what is the automorphism group of $\mathcal{C}$?

The two questions are closely related, since if we are able to find all possible maps between two codes, $\mathcal{C}_1$ and $\mathcal{C}_2$, we can answer both of them (the latter by letting $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$).

Invariants can be very useful in studying the first question.

**Definition 3.2.15** An **invariant** is a property of a code that depends only on the abstract structure, that is, two equivalent (isomorphic) codes necessarily have the same value of an invariant.

**Remark 3.2.16** Two inequivalent (non-isomorphic) codes may or may not have the same value of an invariant.

**Example 3.2.17** The distance distribution is an invariant of codes. This invariant can be used to show that the two unrestricted $(4, 3, 2)_3$ codes $\mathcal{C}_1 = \{0000, 0120, 2121\}$ and $\mathcal{C}_2 = \{1000, 1111, 2112\}$ are inequivalent. Actually, to distinguish these codes an even less sensitive invariant suffices: the number of pairs of codewords with mutual Hamming distance 3 is 0 for $\mathcal{C}_1$ and 1 for $\mathcal{C}_2$.

Since invariants are only occasionally able to provide the right answer to the first question above, alternative techniques are needed for providing the answer in all possible situations. One such technique relies on producing canonical representatives of codes.

**Definition 3.2.18** Let $S$ be a set of possible codes, and let $r : S \to S$ be a map with the properties that (i) $r(\mathcal{C}_1) = r(\mathcal{C}_2)$ if and only if $\mathcal{C}_1$ and $\mathcal{C}_2$ are equivalent (isomorphic) and (ii) $r(\mathcal{C}) = r(r(\mathcal{C}))$. The **canonical representative** (or **canonical form**) of a code $\mathcal{C} \in S$ with respect to this map is $r(\mathcal{C})$.

To test whether two codes are equivalent (isomorphic), it suffices to test their canonical representatives for equality.

**Remark 3.2.19** The number of equivalence (isomorphism) classes that can be handled when comparing canonical representatives is limited by the amount of computer memory available. However, in the context of classifying codes, there are actually methods that do not require any comparison between codes; see [1090].

For particular codes with special properties, algebraic and combinatorial techniques can be used to determine the automorphism group of a code.

**Example 3.2.20** The automorphism group of the $[2^m - 1, 2^m - r - 1, 3]_2$ binary Hamming code is isomorphic to the general linear group $\mathrm{GL}_m(2)$.

**Remark 3.2.21** Determining the automorphism group essentially consists of two parts: finding the set of all automorphisms and proving that the set is complete. The latter task is trivial in cases where the automorphism group is a maximal subgroup of the group of all possible symmetries.

In the general case, algorithmic tools are required to answer the questions above. In the early 1980s, Leon [1217] published an algorithm for computing automorphism groups of linear codes over arbitrary fields, considering monomial equivalence. More recently, Feulner [721] developed an algorithm for computing automorphism groups *and* canonical representatives of linear codes, considering equivalence.

Algorithms similar to those developed by Leon and Feulner could also be developed for unrestricted codes. However, the task of developing such tailored algorithms is rather tedious. An alternative, convenient approach for unrestricted codes and their various subclasses is to map the codes to colored graphs, which can be then be considered in the framework of graph isomorphism. In particular, the `nauty` graph automorphism software [1370] can then be used.

Maps from codes to colored graphs are discussed in [1090, Chap. 3]. For unrestricted codes over $\mathbb{F}_q$ with length $n$ and size $M$, one may consider the following graph with $M + qn$ vertices. Take one vertex for each codeword and a complete graph with $q$ vertices for each coordinate. Insert edges so that the neighbors of a codeword vertex show the values in the respective coordinates. One may color the graph so that no automorphism can map a vertex of one of the two types to the other (although the structure of the graph is such that for nontrivial codes no such maps are possible even if the graph is uncolored).

**Example 3.2.22** For the two codes in Example 3.2.17, one may construct the graphs in Figure 3.1. The leftmost vertex in each triangle corresponds to value 0, and the other two vertices correspond to 1 and 2 in a clockwise manner. Now we can also use graph invariants, such as the distribution of the degrees of the vertices, to see that the graphs are non-isomorphic, which in turn implies that the codes are inequivalent.

**Remark 3.2.23** The number of codewords of linear codes grows exponentially as a function of the dimension of the codes. Hence we do not wish to map individual codewords to vertices of some graph, so an approach similar to that for unrestricted codes is not practical. However, various other approaches have been proposed which partly rely on `nauty`; see [483, 1467, 1608] and [1090, Chap. 7]. Further algorithms for classifying linear codes are presented in [190].

## 3.3 Some Central Classes of Codes

By Definition 1.9.1, the maximum size of error-correcting codes with length $n$ and minimum distance $d$ are given by the functions $A_q(n, d)$ and $B_q(n, d)$ for unrestricted and linear codes, respectively. Most general bounds on these functions, such as those in Section 1.9,