

Codes  
Lattices  
Lie Algebras  
Elliptic Curves  
Modular Forms  
and  
Vertex Operator Algebras

Chris Long

January-March 2000

# Chapter 1

## Introduction

This set of notes will be examining some remarkable structures, including the Hamming codes, the Golay codes, the  $E_8$  root lattice, the Leech lattice, and the moonshine module.

As a quick glance, the dodecahedron yields the binary Golay code, which is used in constructing the Leech lattice, which is used in constructing the moonshine module. This is a typical example of one structure with remarkable symmetries yielding many others.

# Chapter 2

## Error-Correcting Codes

### 2.1 Codes

**Definition 2.1** A code  $C(V)$  is a subset of a vector space  $V$ . If the subset is a vector subspace, the code is said to be **linear**; otherwise the code is said to be **non-linear**. The elements of  $V$  are called **words**, and the elements of  $C$  are called **codewords**.

The vector space  $V$  will also be referred to as the *word space*. The primary area of interest, of course, is when  $V$  is a finite-dimensional vector space over a finite field. But as many of the ideas extend to the more general situation where  $V$  is an arbitrary vector space, it makes sense to make as few assumptions as possible.

**Definition 2.2** A **decoding scheme** for a code  $C(V)$  is an equivalence relation  $\sim$  over  $V$  such that for each equivalence class  $U \subset V$ ,  $|U \cap C| = 1$ . This may also be described by the map  $D : V \rightarrow C$ , where  $D : v \mapsto u$ , with  $u$  being the unique element in  $C$  such that  $u \sim v$ .

Thus, a received word  $v$  gets *decoded* to the codeword  $D(v)$ .

**Definition 2.3** An **error-correcting code** is a code  $C$  together with a decoding scheme  $D$ . A received word  $v$  gets **corrected** to  $D(v)$ .

### 2.2 Codes with Finite Word Spaces

Now assume that  $C(V)$  is a code with a *finite* word space, i.e.  $V$  is a finite vector space (and is therefore a finite-dimensional vector space over a finite field). A linear  $C(V)$  code, where  $V$  is an  $n$ -dimensional vector space and  $C$  is a  $k$ -dimensional vector subspace of  $V$ , will be called an  $(n, k)$  code.

To do much more, we'll need our codes to be linear, so linearity will be assumed in the following.

The most basic model for a situation where we're transferring data between two points, with the possibility of some data corruption during the transmission, is to assume that the

data is composed of codewords, which are vectors in some code space  $C(V)$ . As the data is transferred, each component in the data vectors get corrupted (i.e. changed to an incorrect value) independently with some probability  $p$ . The corruption takes the form of the scalar value of a particular component changing to some other value in our field, the distribution of which we will assume is uniform. Thus, it's likely that our corrupted codeword is no longer in our code space, but is rather a word in the larger word space. What we'd like to find is an intelligent decoding scheme that will allow us to determine which was the most likely original codeword.

**Definition 2.4** *The **Hamming norm** over a finite field is the trivial norm, i.e. for  $x \in F$  define  $|x| = 1$  if  $x \neq 0$ , and  $|x| = 0$  if  $x = 0$ . This may be extended to form the **Hamming norm** over  $V$  by defining  $|\langle v_1, \dots, v_n \rangle| = |v_1| + \dots + |v_n|$ .*

The Hamming norm of a word is also called the **Hamming weight** of a word, or simply the **weight**. As usual, a normed vector space  $V$  becomes a metric space by defining  $d(u, v) = |u - v|$ , and this metric will be referred to as the **Hamming metric**.

For a given code, let  $d$  be the minimum positive weight over all codewords. An  $(n, k, d)$  is an  $(n, k)$  code that has minimum positive weight  $d$ . Note that an  $(n, k, d)$  error-correcting code can correct at most  $\lfloor \frac{d-1}{2} \rfloor$  errors with certainty; in fact, it's an easy exercise that a linear code can correct *exactly* this many errors with certainty.

**Definition 2.5** *A code is **even** if every codeword has even weight. A code is **doubly even** if every codeword has weight a multiple of 4.*

**Definition 2.6** *A **maximum-likelihood decoding scheme** is a decoding scheme that decodes every word to its nearest codeword, with nearest in the sense of the Hamming metric.*

This is an appropriate name, since this codeword is the most likely to be the original codeword, under the assumptions we've made above. Note that such a decoding scheme *requires* that each word must have only one codeword that is the closest to it.

**Example** Consider the  $(5, 3)$  binary code  $C$  with generator matrix

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

The word generator here refers to the rows of  $G$  forming a basis for  $C$ , i.e. every codeword is a linear combination of the rows of  $G$ . Note that  $d = 2$  for this code.

We may clearly assume that the generator matrix for a  $(n, k)$  code has the block matrix structure  $G = (I_k \ A)$ , where  $A$  is a  $k \times (n - k)$  matrix.

**Example** Consider the  $(7, 4)$  binary code  $C$  with generator matrix

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

This is the famous Hamming (7, 4) code, and it has  $d = 3$ , so exactly one error can be corrected with certainty.

**Definition 2.7** A code  $C(V)$  is called **perfect** if all of the vectors in  $V$  are contained in the spheres of radius  $\lfloor \frac{d-1}{2} \rfloor$  centered at the codewords.

The binary Hamming (7, 4, 3) code is a perfect code.

**Definition 2.8** If  $C(V)$  is a code over  $V$ , we define the **dual** or **orthogonal code** to be  $C^\perp = \{v \in V \mid v \cdot w \text{ for all } w \in C\}$ . A code is said to be **self-orthogonal** if  $C \subset C^\perp$ , and a code is said to be **self-dual** if  $C = C^\perp$ .

Note that if  $C$  is  $k$ -dimensional, then  $C^\perp$  is  $(n - k)$ -dimensional.

**Proposition 2.9** A binary code  $C$  is self-orthogonal if and only if the rows of a generator matrix for  $C$  are mutually orthogonal and have even weight. Similarly, a ternary code  $C$  is self-orthogonal if and only if the rows of a generator matrix for  $C$  are mutually orthogonal and have weight a multiple of 3.

**Proof**  $\square$

**Proposition 2.10** A binary code  $C$  is doubly-even and self-orthogonal if and only if the rows of a generator matrix for  $C$  are mutually orthogonal and have weight a multiple of 4.

**Proof**  $\square$

**Example** Extending the above code to a binary (8, 4, 4) code by appending to each row the parity of the row sum, we obtain the generator matrix

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

We may also write  $G$  is the particularly nice block matrix form  $(I_4 \ J_4 - I_4)$ , where  $J_n$  is the  $n \times n$  matrix of all ones. The process of appending this parity check to the generator matrix of  $C$  is called **extending**  $C$ , and resultant code is called the **extended**  $C$ . Note that the extended Hamming code is doubly-even and self-dual, which will become very important later.

**Definition 2.11** The process of creating a new code by removing a column from a generator matrix for a code  $C$  is called **puncturing**  $C$ . The resultant code is called a **punctured**  $C$  code. Note that different punctures may result in inequivalent codes.

**Exercise 1** Show that for the very first example, different punctures may yield inequivalent codes.

**Exercise 2** Do different punctures of the Hamming (8, 4, 4) code yield inequivalent codes?

**Example** Consider the  $12 \times 12$  binary matrix

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

A generator matrix for the (24, 12, 8) binary extended Golay code can then be given in the block matrix form  $(I_{12} \ A)$ . As was the extended Hamming code, the Golay code is doubly-even and self-dual. Puncturing, we obtain the (23, 11, 7) binary Golay code which, like the Hamming code, is perfect. We'll see later that it doesn't matter which column we puncture, as all of the resulting codes will be equivalent.

**Exercise 3** Let  $A$  be the face-adjacency graph for an dodecahedron (and so  $A$  is a  $12 \times 12$  matrix). Show that another generator matrix for the binary extended Golay code is given by the block matrix  $(I_{12} \ J_{12} - A)$ .

**Exercise 4** Does a similar construction for the octahedron yield the binary extended Hamming code?

## 2.3 Weight Enumerator Polynomials

## 2.4 Eine Kleine Invariantentheorie

## 2.5 The Hamming Codes

## 2.6 The Golay Codes

# Chapter 3

## Lattices

# Chapter 4

## Modular Forms

# Chapter 5

## Elliptic Curves

# Bibliography

- [1] Frenkel, Lepowsky, Meurman, Vertex Operator Algebras and the Monster, Academic Press (1988)
- [2] Pless, Introduction to the Theory of Error-Correcting Codes, Wiley-Interscience (1982)