Exercise 8.7.16

a. If Z is an invertible complex matrix, show that Z^H is invertible and that $(Z^H)^{-1} = (Z^{-1})^H$.

b. Show that the inverse of a unitary matrix is again unitary.

c. If U is unitary, show that U^H is unitary.

Exercise 8.7.17 Let Z be an $m \times n$ matrix such that $Z^H Z = I_n$ (for example, Z is a unit column in $\mathbb{C}^n$).

a. Show that $V = ZZ^H$ is hermitian and satisfies $V^2 = V$.

b. Show that $U = I - 2ZZ^H$ is both unitary and hermitian (so $U^{-1} = U^H = U$).

Exercise 8.7.18

a. If N is normal, show that zN is also normal for all complex numbers z.

b. Show that (a) fails if *normal* is replaced by *hermitian*.

Exercise 8.7.19 Show that a real 2×2 normal matrix is either symmetric or has the form $\begin{bmatrix} a & b \\ -b & a \end{bmatrix}$.

Exercise 8.7.20 If A is hermitian, show that all the coefficients of $c_A(x)$ are real numbers.

Exercise 8.7.21

a. If $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, show that $U^{-1}AU$ is not diagonal for any invertible complex matrix U.

b. If $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, show that $U^{-1}AU$ is not upper triangular for any *real* invertible matrix U.

Exercise 8.7.22 If A is any $n \times n$ matrix, show that $U^H AU$ is lower triangular for some unitary matrix U.

Exercise 8.7.23 If A is a 3×3 matrix, show that $A^2 = 0$ if and only if there exists a unitary matrix U such that $U^H AU$ has the form $\begin{bmatrix} 0 & 0 & u \\ 0 & 0 & v \\ 0 & 0 & 0 \end{bmatrix}$ or the form $\begin{bmatrix} 0 & u & v \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

Exercise 8.7.24 If $A^2 = A$, show that $\operatorname{rank} A = \operatorname{tr} A$. [*Hint*: Use Schur's theorem.]

Exercise 8.7.25 Let A be any $n \times n$ complex matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$. Show that $A = P + N$ where $N^n = 0$ and $P = UDU^T$ where U is unitary and $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. [Hint: Schur's theorem]

8.8 An Application to Linear Codes over Finite Fields

For centuries mankind has been using codes to transmit messages. In many cases, for example transmitting financial, medical, or military information, the message is disguised in such a way that it cannot be understood by an intruder who intercepts it, but can be easily "decoded" by the intended receiver. This subject is called *cryptography* and, while intriguing, is not our focus here. Instead, we investigate methods for detecting and correcting errors in the transmission of the message.

The stunning photos of the planet Saturn sent by the space probe are a very good example of how successful these methods can be. These messages are subject to "noise" such as solar interference which causes errors in the message. The signal is received on Earth with errors that must be detected and corrected before the high-quality pictures can be printed. This is done using error-correcting codes. To see how, we first discuss a system of adding and multiplying integers while ignoring multiples of a fixed integer.

Modular Arithmetic

We work in the set $\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \ldots\}$ of **integers**, that is the set of whole numbers. Everyone is familiar with the process of "long division" from arithmetic. For example, we can divide an integer a by 5 and leave a remainder "modulo 5" in the set $\{0, 1, 2, 3, 4\}$. As an illustration

$$19 = 3 \cdot 5 + 4$$

so the remainder of 19 modulo 5 is 4. Similarly, the remainder of 137 modulo 5 is 2 because we have $137 = 27 \cdot 5 + 2$. This works even for negative integers: For example,

$$-17 = (-4) \cdot 5 + 3$$

so the remainder of -17 modulo 5 is 3.

This process is called the **division algorithm**. More formally, let $n \geq 2$ denote an integer. Then every integer a can be written uniquely in the form

$$a = qn + r \quad \text{where } q \text{ and } r \text{ are integers and } 0 \leq r \leq n-1$$

Here q is called the **quotient** of a **modulo** n, and r is called the **remainder** of a **modulo** n. We refer to n as the **modulus**. Thus, if $n = 6$, the fact that $134 = 22 \cdot 6 + 2$ means that 134 has quotient 22 and remainder 2 modulo 6.

Our interest here is in the set of *all* possible remainders modulo n. This set is denoted

$$\mathbb{Z}_n = \{0, 1, 2, 3, \ldots, n-1\}$$

and is called the set of **integers modulo** n. Thus every integer is uniquely represented in $\mathbb{Z}_n$ by its remainder modulo n.

We are going to show how to do arithmetic in $\mathbb{Z}_n$ by adding and multiplying modulo n. That is, we add or multiply two numbers in $\mathbb{Z}_n$ by calculating the usual sum or product in $\mathbb{Z}$ and taking the remainder modulo n. It is proved in books on abstract algebra that the usual laws of arithmetic hold in $\mathbb{Z}_n$ for any modulus $n \geq 2$. This seems remarkable until we remember that these laws are true for ordinary addition and multiplication and all we are doing is reducing modulo n.

To illustrate, consider the case $n = 6$, so that $\mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$. Then $2 + 5 = 1$ in $\mathbb{Z}_6$ because 7 leaves a remainder of 1 when divided by 6. Similarly, $2 \cdot 5 = 4$ in $\mathbb{Z}_6$, while $3 + 5 = 2$, and $3 + 3 = 0$. In this way we can fill in the addition and multiplication tables for $\mathbb{Z}_6$; the result is:

Tables for $\mathbb{Z}_6$

$+$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

$\times$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Calculations in $\mathbb{Z}_6$ are carried out much as in $\mathbb{Z}$. As an illustration, consider the familiar "distributive law" $a(b+c) = ab+ac$ from ordinary arithmetic. This holds for all a, b, and c in $\mathbb{Z}_6$; we verify a particular case:

$$3(5+4) = 3\cdot 5+3\cdot 4 \quad \text{in } \mathbb{Z}_6$$

In fact, the left side is $3(5+4) = 3\cdot 3 = 3$, and the right side is $(3\cdot 5)+(3\cdot 4) = 3+0 = 3$ too. Hence doing arithmetic in $\mathbb{Z}_6$ is familiar. However, there are differences. For example, $3\cdot 4 = 0$ in $\mathbb{Z}_6$, in contrast to the fact that $a\cdot b = 0$ in $\mathbb{Z}$ can only happen when either $a = 0$ or $b = 0$. Similarly, $3^2 = 3$ in $\mathbb{Z}_6$, unlike $\mathbb{Z}$.

Note that we will make statements like $-30 = 19$ in $\mathbb{Z}_7$; it means that -30 and 19 leave the same remainder 5 when divided by 7, and so are equal in $\mathbb{Z}_7$ because they both equal 5. In general, if $n \geq 2$ is any modulus, the operative fact is that

$$a = b \text{ in } \mathbb{Z}_n \quad \text{if and only if} \quad a-b \text{ is a multiple of } n$$

In this case we say that a and b are **equal modulo** n, and write $a = b(\bmod n)$.

Arithmetic in $\mathbb{Z}_n$ is, in a sense, simpler than that for the integers. For example, consider negatives. Given the element 8 in $\mathbb{Z}_{17}$, what is -8? The answer lies in the observation that $8+9 = 0$ in $\mathbb{Z}_{17}$, so $-8 = 9$ (and $-9 = 8$). In the same way, finding negatives is not difficult in $\mathbb{Z}_n$ for any modulus n.

Finite Fields

In our study of linear algebra so far the scalars have been real (possibly complex) numbers. The set $\mathbb{R}$ of real numbers has the property that it is closed under addition and multiplication, that the usual laws of arithmetic hold, and that every nonzero real number has an inverse in $\mathbb{R}$. Such a system is called a **field**. Hence the real numbers $\mathbb{R}$ form a field, as does the set $\mathbb{C}$ of complex numbers. Another example is the set $\mathbb{Q}$ of all rational numbers (fractions); however the set $\mathbb{Z}$ of integers is *not* a field—for example, 2 has no inverse *in* the set $\mathbb{Z}$ because $2\cdot x = 1$ has no solution x in $\mathbb{Z}$.

Our motivation for isolating the concept of a field is that nearly everything we have done remains valid if the scalars are restricted to some field: The gaussian algorithm can be used to solve systems of linear equations with coefficients in the field; a square matrix with entries from the field is invertible if and only if its determinant is nonzero; the matrix inversion algorithm works in the same way; and so on. The reason is that the field has all the properties used in the proofs of these results for the field $\mathbb{R}$, so all the theorems remain valid.

It turns out that there are *finite* fields—that is, finite sets that satisfy the usual laws of arithmetic and in which every nonzero element a has an **inverse**, that is an element b in the field such that $ab = 1$. If $n \geq 2$ is an integer, the modular system $\mathbb{Z}_n$ certainly satisfies the basic laws of arithmetic, but it need not be a field. For example we have $2\cdot 3 = 0$ in $\mathbb{Z}_6$ so 3 has no inverse in $\mathbb{Z}_6$ (if $3a = 1$ then $2 = 2\cdot 1 = 2(3a) = 0a = 0$ in $\mathbb{Z}_6$, a contradiction). The problem is that $6 = 2\cdot 3$ can be properly factored in $\mathbb{Z}$.

An integer $p \geq 2$ is called a **prime** if p *cannot* be factored as $p = ab$ where a and b are positive integers and neither a nor b equals 1. Thus the first few primes are 2, 3, 5, 7, 11, 13, 17, ... If $n \geq 2$ is not a prime and $n = ab$ where $2 \leq a$, $b \leq n-1$, then $ab = 0$ in $\mathbb{Z}_n$ and it follows (as above in the case $n = 6$) that b cannot have an inverse in $\mathbb{Z}_n$, and hence that $\mathbb{Z}_n$ is not a field. In other words, if $\mathbb{Z}_n$ is a field, then n must be a prime. Surprisingly, the converse is true:

Theorem 8.8.1

If p is a prime, then $\mathbb{Z}_p$ is a field using addition and multiplication modulo p.

The proof can be found in books on abstract algebra.[18] If p is a prime, the field $\mathbb{Z}_p$ is called the **field of integers modulo** p.

For example, consider the case $n = 5$. Then $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$ and the addition and multiplication tables are:

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\times$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Hence 1 and 4 are self-inverse in $\mathbb{Z}_5$, and 2 and 3 are inverses of each other, so $\mathbb{Z}_5$ is indeed a field. Here is another important example.

Example 8.8.1

If $p = 2$, then $\mathbb{Z}_2 = \{0, 1\}$ is a field with addition and multiplication modulo 2 given by the tables

$+$	0	1
0	0	1
1	1	0

and

$\times$	0	1
0	0	0
1	0	1

This is binary arithmetic, the basic algebra of computers.

While it is routine to find negatives of elements of $\mathbb{Z}_p$, it is a bit more difficult to find inverses in $\mathbb{Z}_p$. For example, how does one find 14^{-1} in $\mathbb{Z}_{17}$? Since we want $14^{-1} \cdot 14 = 1$ in $\mathbb{Z}_{17}$, we are looking for an integer a with the property that $a \cdot 14 = 1$ modulo 17. Of course we can try all possibilities in $\mathbb{Z}_{17}$ (there are only 17 of them!), and the result is $a = 11$ (verify). However this method is of little use for large primes p, and it is a comfort to know that there is a systematic procedure (called the **euclidean algorithm**) for finding inverses in $\mathbb{Z}_p$ for any prime p. Furthermore, this algorithm is easy to program for a computer. To illustrate the method, let us once again find the inverse of 14 in $\mathbb{Z}_{17}$.

Example 8.8.2

Find the inverse of 14 in $\mathbb{Z}_{17}$.

Solution. The idea is to first divide $p = 17$ by 14:

$$17 = 1 \cdot 14 + 3$$

Now divide (the previous divisor) 14 by the new remainder 3 to get

$$14 = 4 \cdot 3 + 2$$

[18] See, for example, W. Keith Nicholson, *Introduction to Abstract Algebra*, 4th ed., (New York: Wiley, 2012).

and then divide (the previous divisor) 3 by the new remainder 2 to get

$$3 = 1 \cdot 2 + 1$$

It is a theorem of number theory that, because 17 is a prime, this procedure will *always* lead to a remainder of 1. At this point we eliminate remainders in these equations from the bottom up:

$$\begin{aligned} 1 &= 3 - 1 \cdot 2 && \text{since } 3 = 1 \cdot 2 + 1 \\ &= 3 - 1 \cdot (14 - 4 \cdot 3) = 5 \cdot 3 - 1 \cdot 14 && \text{since } 2 = 14 - 4 \cdot 3 \\ &= 5 \cdot (17 - 1 \cdot 14) - 1 \cdot 14 = 5 \cdot 17 - 6 \cdot 14 && \text{since } 3 = 17 - 1 \cdot 14 \end{aligned}$$

Hence $(-6) \cdot 14 = 1$ in $\mathbb{Z}_{17}$, that is, $11 \cdot 14 = 1$. So $14^{-1} = 11$ in $\mathbb{Z}_{17}$.

As mentioned above, nearly everything we have done with matrices over the field of real numbers can be done in the same way for matrices with entries from $\mathbb{Z}_p$. We illustrate this with one example. Again the reader is referred to books on abstract algebra.

Example 8.8.3

Determine if the matrix $A = \begin{bmatrix} 1 & 4 \\ 6 & 5 \end{bmatrix}$ from $\mathbb{Z}_7$ is invertible and, if so, find its inverse.

Solution. Working in $\mathbb{Z}_7$ we have $\det A = 1 \cdot 5 - 6 \cdot 4 = 5 - 3 = 2 \neq 0$ in $\mathbb{Z}_7$, so A is invertible. Hence Example 2.4.4 gives $A^{-1} = 2^{-1} \begin{bmatrix} 5 & -4 \\ -6 & 1 \end{bmatrix}$. Note that $2^{-1} = 4$ in $\mathbb{Z}_7$ (because $2 \cdot 4 = 1$ in $\mathbb{Z}_7$). Note also that $-4 = 3$ and $-6 = 1$ in $\mathbb{Z}_7$, so finally $A^{-1} = 4 \begin{bmatrix} 5 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 4 & 4 \end{bmatrix}$. The reader can verify that indeed $\begin{bmatrix} 1 & 4 \\ 6 & 5 \end{bmatrix} \begin{bmatrix} 6 & 5 \\ 4 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ in $\mathbb{Z}_7$.

While we shall not use them, there are finite fields other than $\mathbb{Z}_p$ for the various primes p. Surprisingly, for every prime p and every integer $n \geq 1$, there *exists* a field with exactly p^n elements, and this field is *unique*.[19] It is called the **Galois field** of order p^n, and is denoted $GF(p^n)$.

[19] See, for example, W. K. Nicholson, *Introduction to Abstract Algebra*, 4th ed., (New York: Wiley, 2012).

Error Correcting Codes

Coding theory is concerned with the transmission of information over a *channel* that is affected by *noise*. The noise causes errors, so the aim of the theory is to find ways to detect such errors and correct at least some of them. General coding theory originated with the work of Claude Shannon (1916–2001) who showed that information can be transmitted at near optimal rates with arbitrarily small chance of error.

Let F denote a finite field and, if $n \geq 1$, let

$$F^n \text{ denote the } F\text{-vector space of } 1 \times n \text{ row matrices over } F$$

with the usual componentwise addition and scalar multiplication. In this context, the rows in F^n are called **words** (or n-**words**) and, as the name implies, will be written as $[a\ b\ c\ d] = abcd$. The individual components of a word are called its **digits**. A nonempty subset C of F^n is called a **code** (or an n-**code**), and the elements in C are called **code words**. If $F = \mathbb{Z}_2$, these are called **binary** codes.

If a code word $\mathbf{w}$ is transmitted and an error occurs, the resulting word $\mathbf{v}$ is decoded as the code word "closest" to $\mathbf{v}$ in F^n. To make sense of what "closest" means, we need a distance function on F^n analogous to that in $\mathbb{R}^n$ (see Theorem 5.3.3). The usual definition in $\mathbb{R}^n$ does not work in this situation. For example, if $\mathbf{w} = 1111$ in $(\mathbb{Z}_2)^4$ then the square of the distance of $\mathbf{w}$ from $\mathbf{0}$ is

$$(1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 = 0$$

even though $\mathbf{w} \neq \mathbf{0}$.

However there is a satisfactory notion of distance in F^n due to Richard Hamming (1915–1998). Given a word $\mathbf{w} = a_1a_2\cdots a_n$ in F^n, we first define the **Hamming weight** $wt(\mathbf{w})$ to be the number of nonzero digits in $\mathbf{w}$:

$$wt(\mathbf{w}) = wt(a_1a_2\cdots a_n) = |\{i \mid a_i \neq 0\}|$$

Clearly, $0 \leq wt(\mathbf{w}) \leq n$ for every word $\mathbf{w}$ in F^n. Given another word $\mathbf{v} = b_1b_2\cdots b_n$ in F^n, the **Hamming distance** $d(\mathbf{v},\ \mathbf{w})$ between $\mathbf{v}$ and $\mathbf{w}$ is defined by

$$d(\mathbf{v},\ \mathbf{w}) = wt(\mathbf{v} - \mathbf{w}) = |\{i \mid b_i \neq a_i\}|$$

In other words, $d(\mathbf{v},\ \mathbf{w})$ is the number of places at which the digits of $\mathbf{v}$ and $\mathbf{w}$ differ. The next result justifies using the term *distance* for this function d.

Theorem 8.8.2

Let $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ denote words in F^n. Then:

1. $d(\mathbf{v},\ \mathbf{w}) \geq 0$.
2. $d(\mathbf{v},\ \mathbf{w}) = 0$ *if and only if* $\mathbf{v} = \mathbf{w}$.
3. $d(\mathbf{v},\ \mathbf{w}) = d(\mathbf{w},\ \mathbf{v})$.
4. $d(\mathbf{v},\ \mathbf{w}) \leq d(\mathbf{v},\ \mathbf{u}) + d(\mathbf{u},\ \mathbf{w})$

Proof. (1) and (3) are clear, and (2) follows because $wt(\mathbf{v}) = 0$ if and only if $\mathbf{v} = \mathbf{0}$. To prove (4), write $\mathbf{x} = \mathbf{v} - \mathbf{u}$ and $\mathbf{y} = \mathbf{u} - \mathbf{w}$. Then (4) reads $wt(\mathbf{x}+\mathbf{y}) \leq wt(\mathbf{x}) + wt(\mathbf{y})$. If $\mathbf{x} = a_1 a_2 \cdots a_n$ and $\mathbf{y} = b_1 b_2 \cdots b_n$, this follows because $a_i + b_i \neq 0$ implies that either $a_i \neq 0$ or $b_i \neq 0$. □

Given a word $\mathbf{w}$ in F^n and a real number $r > 0$, define the **ball** $B_r(\mathbf{w})$ of radius r (or simply the r-**ball**) about $\mathbf{w}$ as follows:

$$B_r(\mathbf{w}) = \{\mathbf{x} \in F^n \mid d(\mathbf{w}, \mathbf{x}) \leq r\}$$

Using this we can describe one of the most useful decoding methods.

Nearest Neighbour Decoding

Let C be an n-code, and suppose a word $\mathbf{v}$ is transmitted and $\mathbf{w}$ is received. Then $\mathbf{w}$ is decoded as the code word in C closest to it. (If there is a tie, choose arbitrarily.)

Using this method, we can describe how to construct a code C that can detect (or correct) t errors. Suppose a code word $\mathbf{c}$ is transmitted and a word $\mathbf{w}$ is received with s errors where $1 \leq s \leq t$. Then s is the number of places at which the $\mathbf{c}$- and $\mathbf{w}$-digits differ, that is, $s = d(\mathbf{c}, \mathbf{w})$. Hence $B_t(\mathbf{c})$ consists of all possible received words where at most t errors have occurred.

Assume first that C has the property that no code word lies in the t-ball of another code word. Because $\mathbf{w}$ is in $B_t(\mathbf{c})$ and $\mathbf{w} \neq \mathbf{c}$, this means that $\mathbf{w}$ is not a code word and the error has been detected. If we strengthen the assumption on C to require that the t-balls about code words are pairwise disjoint, then $\mathbf{w}$ belongs to a unique ball (the one about $\mathbf{c}$), and so $\mathbf{w}$ will be correctly decoded as $\mathbf{c}$.

To describe when this happens, let C be an n-code. The **minimum distance** d of C is defined to be the smallest distance between two distinct code words in C; that is,

$$d = \min\{d(\mathbf{v}, \mathbf{w}) \mid \mathbf{v} \text{ and } \mathbf{w} \text{ in } C; \mathbf{v} \neq \mathbf{w}\}$$

Theorem 8.8.3

Let C be an n-code with minimum distance d. Assume that nearest neighbour decoding is used. Then:

1. *If $t < d$, then C can detect t errors.*[20]
2. *If $2t < d$, then C can correct t errors.*

Proof.

1. Let $\mathbf{c}$ be a code word in C. If $\mathbf{w} \in B_t(\mathbf{c})$, then $d(\mathbf{w}, \mathbf{c}) \leq t < d$ by hypothesis. Thus the t-ball $B_t(\mathbf{c})$ contains no other code word, so C can detect t errors by the preceding discussion.
2. If $2t < d$, it suffices (again by the preceding discussion) to show that the t-balls about distinct code words are pairwise disjoint. But if $\mathbf{c} \neq \mathbf{c}'$ are code words in C and $\mathbf{w}$ is in $B_t(\mathbf{c}') \cap B_t(\mathbf{c})$, then Theorem 8.8.2 gives
$$d(\mathbf{c}, \mathbf{c}') \leq d(\mathbf{c}, \mathbf{w}) + d(\mathbf{w}, \mathbf{c}') \leq t + t = 2t < d$$
by hypothesis, contradicting the minimality of d. □

[20] We say that C detects (corrects) t errors if C can detect (or correct) t or *fewer* errors.

Example 8.8.4

If $F = \mathbb{Z}_3 = \{0, 1, 2\}$, the 6-code $\{111111, 111222, 222111\}$ has minimum distance 3 and so can detect 2 errors and correct 1 error.

Let $\mathbf{c}$ be any word in F^n. A word $\mathbf{w}$ satisfies $d(\mathbf{w}, \mathbf{c}) = r$ if and only if $\mathbf{w}$ and $\mathbf{c}$ differ in exactly r digits. If $|F| = q$, there are exactly $\binom{n}{r}(q-1)^r$ such words where $\binom{n}{r}$ is the binomial coefficient. Indeed, choose the r places where they differ in $\binom{n}{r}$ ways, and then fill those places in $\mathbf{w}$ in $(q-1)^r$ ways. It follows that the number of words in the t-ball about $\mathbf{c}$ is

$$|B_t(\mathbf{c})| = \binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t = \sum_{i=0}^{t}\binom{n}{i}(q-1)^i$$

This leads to a useful bound on the size of error-correcting codes.

Theorem 8.8.4: Hamming Bound

Let C be an n-code over a field F that can correct t errors using nearest neighbour decoding. If $|F| = q$, then

$$|C| \leq \frac{q^n}{\sum_{i=0}^{t}\binom{n}{i}(q-1)^i}$$

Proof. Write $k = \sum_{i=0}^{t}\binom{n}{i}(q-1)^i$. The t-balls centred at distinct code words each contain k words, and there are $|C|$ of them. Moreover they are pairwise disjoint because the code corrects t errors (see the discussion preceding Theorem 8.8.3). Hence they contain $k \cdot |C|$ distinct words, and so $k \cdot |C| \leq |F^n| = q^n$, proving the theorem. □

A code is called **perfect** if there is equality in the Hamming bound; equivalently, if every word in F^n lies in exactly one t-ball about a code word. For example, if $F = \mathbb{Z}_2$, $n = 3$, and $t = 1$, then $q = 2$ and $\binom{3}{0} + \binom{3}{1} = 4$, so the Hamming bound is $\frac{2^3}{4} = 2$. The 3-code $C = \{000, 111\}$ has minimum distance 3 and so can correct 1 error by Theorem 8.8.3. Hence C is perfect.

Linear Codes

Up to this point we have been regarding *any* nonempty subset of the F-vector space F^n as a code. However many important codes are actually subspaces. A subspace $C \subseteq F^n$ of dimension $k \geq 1$ over F is called an (n, k)**-linear code**, or simply an (n, k)**-code**. We do not regard the zero subspace (that is, $k = 0$) as a code.

Example 8.8.5

If $F = \mathbb{Z}_2$ and $n \geq 2$, the n-**parity-check code** is constructed as follows: An extra digit is added to each word in F^{n-1} to make the number of 1s in the resulting word even (we say such words have **even parity**). The resulting $(n, n-1)$-code is linear because the sum of two words of even parity again has even parity.

Many of the properties of general codes take a simpler form for linear codes. The following result gives a much easier way to find the minimal distance of a linear code, and sharpens the results in Theorem 8.8.3.

Theorem 8.8.5

Let C be an $(n,\ k)$-code with minimum distance d over a finite field F, and use nearest neighbour decoding.

1. $d = \min\{wt(\mathbf{w}) \mid \mathbf{0} \neq \mathbf{w} \in C\}$.
2. *C can detect $t \geq 1$ errors if and only if $t < d$.*
3. *C can correct $t \geq 1$ errors if and only if $2t < d$.*
4. *If C can correct $t \geq 1$ errors and $|F| = q$, then*

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t \leq q^{n-k}$$

Proof.

1. Write $d' = \min\{wt(\mathbf{w}) \mid \mathbf{0} \neq \mathbf{w} \text{ in } C\}$. If $\mathbf{v} \neq \mathbf{w}$ are words in C, then $d(\mathbf{v},\ \mathbf{w}) = wt(\mathbf{v} - \mathbf{w}) \geq d'$ because $\mathbf{v} - \mathbf{w}$ is in the subspace C. Hence $d \geq d'$. Conversely, given $\mathbf{w} \neq \mathbf{0}$ in C then, since $\mathbf{0}$ is in C, we have $wt(\mathbf{w}) = d(\mathbf{w},\ \mathbf{0}) \geq d$ by the definition of d. Hence $d' \geq d$ and (1) is proved.

2. Assume that C can detect t errors. Given $\mathbf{w} \neq \mathbf{0}$ in C, the t-ball $B_t(\mathbf{w})$ about $\mathbf{w}$ contains no other code word (see the discussion preceding Theorem 8.8.3). In particular, it does not contain the code word $\mathbf{0}$, so $t < d(\mathbf{w},\ \mathbf{0}) = wt(\mathbf{w})$. Hence $t < d$ by (1). The converse is part of Theorem 8.8.3.

3. We require a result of interest in itself.

   Claim. Suppose $\mathbf{c}$ in C has $wt(\mathbf{c}) \leq 2t$. Then $B_t(\mathbf{0}) \cap B_t(\mathbf{c})$ is nonempty.

   Proof. If $wt(\mathbf{c}) \leq t$, then $\mathbf{c}$ itself is in $B_t(\mathbf{0}) \cap B_t(\mathbf{c})$. So assume $t < wt(\mathbf{c}) \leq 2t$. Then $\mathbf{c}$ has more than t nonzero digits, so we can form a new word $\mathbf{w}$ by changing exactly t of these nonzero digits to zero. Then $d(\mathbf{w},\ \mathbf{c}) = t$, so $\mathbf{w}$ is in $B_t(\mathbf{c})$. But $wt(\mathbf{w}) = wt(\mathbf{c}) - t \leq t$, so $\mathbf{w}$ is also in $B_t(\mathbf{0})$. Hence $\mathbf{w}$ is in $B_t(\mathbf{0}) \cap B_t(\mathbf{c})$, proving the Claim.

   If C corrects t errors, the t-balls about code words are pairwise disjoint (see the discussion preceding Theorem 8.8.3). Hence the claim shows that $wt(\mathbf{c}) > 2t$ for all $\mathbf{c} \neq \mathbf{0}$ in C, from which $d > 2t$ by (1). The other inequality comes from Theorem 8.8.3.

4. We have $|C| = q^k$ because $\dim_F C = k$, so this assertion restates Theorem 8.8.4. □

Example 8.8.6

If $F = \mathbb{Z}_2$, then

$$C = \{0000000,\ 0101010,\ 1010101,\ 1110000,\ 1011010,\ 0100101,\ 0001111,\ 1111111\}$$

is a $(7,\ 3)$-code; in fact $C = \text{span}\{0101010,\ 1010101,\ 1110000\}$. The minimum distance for C is 3, the minimum weight of a nonzero word in C.

Matrix Generators

Given a linear n-code C over a finite field F, the way encoding works in practice is as follows. A message stream is blocked off into segments of length $k \leq n$ called **messages**. Each message $\mathbf{u}$ in F^k is encoded as a code word, the code word is transmitted, the receiver decodes the received word as the nearest code word, and then re-creates the original message. A fast and convenient method is needed to encode the incoming messages, to decode the received word after transmission (with or without error), and finally to retrieve messages from code words. All this can be achieved for any linear code using matrix multiplication.

Let G denote a $k \times n$ matrix over a finite field F, and encode each message $\mathbf{u}$ in F^k as the word $\mathbf{u}G$ in F^n using matrix multiplication (thinking of words as rows). This amounts to saying that the set of code words is the subspace $C = \{\mathbf{u}G \mid \mathbf{u} \text{ in } F^k\}$ of F^n. This subspace need not have dimension k for every $k \times n$ matrix G. But, if $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k\}$ is the standard basis of F^k, then $\mathbf{e}_i G$ is row i of G for each I and $\{\mathbf{e}_1 G, \mathbf{e}_2 G, \ldots, \mathbf{e}_k G\}$ spans C. Hence $\dim C = k$ if and only if the rows of G are independent in F^n, and these matrices turn out to be exactly the ones we need. For reference, we state their main properties in Lemma 8.8.1 below (see Theorem 5.4.4).

Lemma 8.8.1

The following are equivalent for a $k \times n$ matrix G over a finite field F:

1. $\operatorname{rank} G = k$.
2. *The columns of G span F^k.*
3. *The rows of G are independent in F^n.*
4. *The system $GX = B$ is consistent for every column B in $\mathbb{R}^k$.*
5. *$GK = I_k$ for some $n \times k$ matrix K.*

Proof. (1) $\Rightarrow$ (2). This is because $\dim(\operatorname{col} G) = k$ by (1).

(2) $\Rightarrow$ (4). $G \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T = x_1\mathbf{c}_1 + \cdots + x_n\mathbf{c}_n$ where $\mathbf{c}_j$ is column j of G.

(4) $\Rightarrow$ (5). $G \begin{bmatrix} \mathbf{k}_1 & \cdots & \mathbf{k}_k \end{bmatrix} = \begin{bmatrix} G\mathbf{k}_1 & \cdots & G\mathbf{k}_k \end{bmatrix}$ for columns $\mathbf{k}_j$.

(5) $\Rightarrow$ (3). If $a_1R_1 + \cdots + a_kR_k = 0$ where R_i is row i of G, then $\begin{bmatrix} a_1 & \cdots & a_k \end{bmatrix} G = 0$, so by (5), $\begin{bmatrix} a_1 & \cdots & a_k \end{bmatrix} = 0$. Hence each $a_i = 0$, proving (3).

(3) $\Rightarrow$ (1). $\operatorname{rank} G = \dim(\operatorname{row} G) = k$ by (3). □

Note that Theorem 5.4.4 asserts that, over the real field $\mathbb{R}$, the properties in Lemma 8.8.1 hold if and only if GG^T is invertible. But this need not be true in general. For example, if $F = \mathbb{Z}_2$ and $G = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$, then $GG^T = 0$. The reason is that the dot product $\mathbf{w} \cdot \mathbf{w}$ can be zero for $\mathbf{w}$ in F^n even if $\mathbf{w} \neq \mathbf{0}$. However, even though GG^T is not invertible, we do have $GK = I_2$ for some 4×2 matrix K over F as Lemma 8.8.1 asserts (in fact, $K = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^T$ is one such matrix).

Let $C \subseteq F^n$ be an $(n,\ k)$-code over a finite field F. If $\{\mathbf{w}_1,\ \ldots,\ \mathbf{w}_k\}$ is a basis of C, let $G = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_k \end{bmatrix}$ be the $k \times n$ matrix with the $\mathbf{w}_i$ as its rows. Let $\{\mathbf{e}_1,\ \ldots,\ \mathbf{e}_k\}$ is the standard basis of F^k regarded as rows. Then $\mathbf{w}_i = \mathbf{e}_i G$ for each i, so $C = \text{span}\{\mathbf{w}_1,\ \ldots,\ \mathbf{w}_k\} = \text{span}\{\mathbf{e}_1 G,\ \ldots,\ \mathbf{e}_k G\}$. It follows (verify) that

$$C = \{\mathbf{u}G \mid \mathbf{u} \text{ in } F^k\}$$

Because of this, the $k \times n$ matrix G is called a **generator** of the code C, and G has rank k by Lemma 8.8.1 because its rows $\mathbf{w}_i$ are independent.

In fact, every linear code C in F^n has a generator of a simple, convenient form. If G is a generator matrix for C, let R be the reduced row-echelon form of G. We claim that C is also generated by R. Since $G \to R$ by row operations, Theorem 2.5.1 shows that these same row operations $\begin{bmatrix} G & I_k \end{bmatrix} \to \begin{bmatrix} R & W \end{bmatrix}$, performed on $\begin{bmatrix} G & I_k \end{bmatrix}$, produce an invertible $k \times k$ matrix W such that $R = WG$. Then $C = \{\mathbf{u}R \mid \mathbf{u} \text{ in } F^k\}$. [In fact, if $\mathbf{u}$ is in F^k, then $\mathbf{u}G = \mathbf{u}_1 R$ where $\mathbf{u}_1 = \mathbf{u}W^{-1}$ is in F^k, and $\mathbf{u}R = \mathbf{u}_2 G$ where $\mathbf{u}_2 = \mathbf{u}W$ is in F^k]. Thus R is a generator of C, so we may assume that G is in reduced row-echelon form.

In that case, G has no row of zeros (since $\text{rank } G = k$) and so contains all the columns of I_k. Hence a series of *column* interchanges will carry G to the block form $G'' = \begin{bmatrix} I_k & A \end{bmatrix}$ for some $k \times (n-k)$ matrix A. Hence the code $C'' = \{\mathbf{u}G'' \mid \mathbf{u} \text{ in } F^k\}$ is essentially the same as C; the code words in C'' are obtained from those in C by a series of column interchanges. Hence if C is a linear $(n,\ k)$-code, we may (and shall) assume that the generator matrix G has the form

$$G = \begin{bmatrix} I_k & A \end{bmatrix} \quad \text{for some} \quad k \times (n-k) \text{ matrix } A$$

Such a matrix is called a **standard generator**, or a **systematic generator**, for the code C. In this case, if $\mathbf{u}$ is a message word in F^k, the first k digits of the encoded word $\mathbf{u}G$ are just the first k digits of $\mathbf{u}$, so retrieval of $\mathbf{u}$ from $\mathbf{u}G$ is very simple indeed. The last $n-k$ digits of $\mathbf{u}G$ are called **parity digits**.

Parity-Check Matrices

We begin with an important theorem about matrices over a finite field.

Theorem 8.8.6

Let F be a finite field, let G be a $k \times n$ matrix of rank k, let H be an $(n-k) \times n$ matrix of rank $n-k$, and let $C = \{\mathbf{u}G \mid \mathbf{u}$ in $F^k\}$ and $D = \{\mathbf{v}H \mid \mathbf{V}$ in $F^{n-k}\}$ be the codes they generate. Then the following conditions are equivalent:

1. $GH^T = 0$.
2. $HG^T = 0$.
3. $C = \{\mathbf{w}$ *in* $F^n \mid \mathbf{w}H^T = \mathbf{0}\}$.
4. $D = \{\mathbf{w}$ *in* $F^n \mid \mathbf{w}G^T = \mathbf{0}\}$.

Proof. First, (1) $\Leftrightarrow$ (2) holds because HG^T and GH^T are transposes of each other.

(1) $\Rightarrow$ (3) Consider the linear transformation $T : F^n \to F^{n-k}$ defined by $T(\mathbf{w}) = \mathbf{w}H^T$ for all $\mathbf{w}$ in F^n. To prove (3) we must show that $C = \ker T$. We have $C \subseteq \ker T$ by (1) because $T(\mathbf{u}G) = \mathbf{u}GH^T = \mathbf{0}$ for all $\mathbf{u}$ in F^k. Since $\dim C = \operatorname{rank} G = k$, it is enough (by Theorem 6.4.2) to show $\dim(\ker T) = k$. However the dimension theorem (Theorem 7.2.4) shows that $\dim(\ker T) = n - \dim(\operatorname{im} T)$, so it is enough to show that $\dim(\operatorname{im} T) = n - k$. But if $R_1, \ldots, R_n$ are the rows of H^T, then block multiplication gives

$$\operatorname{im} T = \{\mathbf{w}H^T \mid \mathbf{w} \text{ in } \mathbb{R}^n\} = \operatorname{span}\{R_1, \ldots, R_n\} = \operatorname{row}(H^T)$$

Hence $\dim(\operatorname{im} T) = \operatorname{rank}(H^T) = \operatorname{rank} H = n - k$, as required. This proves (3).

(3) $\Rightarrow$ (1) If $\mathbf{u}$ is in F^k, then $\mathbf{u}G$ is in C so, by (3), $\mathbf{u}(GH^T) = (\mathbf{u}G)H^T = \mathbf{0}$. Since $\mathbf{u}$ is arbitrary in F^k, it follows that $GH^T = 0$.

(2) $\Leftrightarrow$ (4) The proof is analogous to (1) $\Leftrightarrow$ (3). □

The relationship between the codes C and D in Theorem 8.8.6 will be characterized in another way in the next subsection.

If C is an (n, k)-code, an $(n-k) \times n$ matrix H is called a **parity-check matrix** for C if $C = \{\mathbf{w} \mid \mathbf{w}H^T = \mathbf{0}\}$ as in Theorem 8.8.6. Such matrices are easy to find for a given code C. If $G = \begin{bmatrix} I_k & A \end{bmatrix}$ is a standard generator for C where A is $k \times (n-k)$, the $(n-k) \times n$ matrix

$$H = \begin{bmatrix} -A^T & I_{n-k} \end{bmatrix}$$

is a parity-check matrix for C. Indeed, $\operatorname{rank} H = n - k$ because the rows of H are independent (due to the presence of I_{n-k}), and

$$GH^T = \begin{bmatrix} I_k & A \end{bmatrix} \begin{bmatrix} -A \\ I_{n-k} \end{bmatrix} = -A + A = 0$$

by block multiplication. Hence H is a parity-check matrix for C and we have $C = \{\mathbf{w} \text{ in } F^n \mid \mathbf{w}H^T = \mathbf{0}\}$. Since $\mathbf{w}H^T$ and $H\mathbf{w}^T$ are transposes of each other, this shows that C can be characterized as follows:

$$C = \{\mathbf{w} \text{ in } F^n \mid H\mathbf{w}^T = \mathbf{0}\}$$

by Theorem 8.8.6.

This is useful in decoding. The reason is that decoding is done as follows: If a code word $\mathbf{c}$ is transmitted and $\mathbf{v}$ is received, then $\mathbf{z} = \mathbf{v} - \mathbf{c}$ is called the **error**. Since $H\mathbf{c}^T = \mathbf{0}$, we have $H\mathbf{z}^T = H\mathbf{v}^T$ and this word

$$\mathbf{s} = H\mathbf{z}^T = H\mathbf{v}^T$$

is called the **syndrome**. The receiver knows $\mathbf{v}$ and $\mathbf{s} = H\mathbf{v}^T$, and wants to recover $\mathbf{c}$. Since $\mathbf{c} = \mathbf{v} - \mathbf{z}$, it is enough to find $\mathbf{z}$. But the possibilities for $\mathbf{z}$ are the solutions of the linear system

$$H\mathbf{z}^T = \mathbf{s}$$

where $\mathbf{s}$ is known. Now recall that Theorem 2.2.3 shows that these solutions have the form $\mathbf{z} = \mathbf{x} + \mathbf{s}$ where $\mathbf{x}$ is any solution of the homogeneous system $H\mathbf{x}^T = \mathbf{0}$, that is, $\mathbf{x}$ is any word in C (by Lemma 8.8.1). In other words, the errors $\mathbf{z}$ are the elements of the set

$$C + \mathbf{s} = \{\mathbf{c} + \mathbf{s} \mid \mathbf{c} \text{ in } C\}$$

The set $C + \mathbf{s}$ is called a **coset** of C. Let $|F| = q$. Since $|C + \mathbf{s}| = |C| = q^{n-k}$ the search for $\mathbf{z}$ is reduced from q^n possibilities in F^n to q^{n-k} possibilities in $C + \mathbf{s}$. This is called **syndrome decoding**, and various

methods for improving efficiency and accuracy have been devised. The reader is referred to books on coding for more details.[21]

Orthogonal Codes

Let F be a finite field. Given two words $\mathbf{v} = a_1a_2\cdots a_n$ and $\mathbf{w} = b_1b_2\cdots b_n$ in F^n, the dot product $\mathbf{v}\cdot\mathbf{w}$ is defined (as in $\mathbb{R}^n$) by

$$\mathbf{v}\cdot\mathbf{w} = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$

Note that $\mathbf{v}\cdot\mathbf{w}$ is an element of F, and it can be computed as a matrix product: $\mathbf{v}\cdot\mathbf{w} = \mathbf{v}\mathbf{w}^T$.

If $C \subseteq F^n$ is an $(n,\ k)$-code, the **orthogonal complement** $C^\perp$ is defined as in $\mathbb{R}^n$:

$$C^\perp = \{\mathbf{v} \text{ in } F^n \mid \mathbf{v}\cdot\mathbf{c} = 0 \text{ for all } \mathbf{c} \text{ in } C\}$$

This is easily seen to be a subspace of F^n, and it turns out to be an $(n,\ n-k)$-code. This follows when $F = \mathbb{R}$ because we showed (in the projection theorem) that $n = \dim U^\perp + \dim U$ for any subspace U of $\mathbb{R}^n$. However the proofs break down for a finite field F because the dot product in F^n has the property that $\mathbf{w}\cdot\mathbf{w} = 0$ can happen even if $\mathbf{w} \neq \mathbf{0}$. Nonetheless, the result remains valid.

Theorem 8.8.7

Let C be an $(n,\ k)$-code over a finite field F, let $G = \begin{bmatrix} I_k & A \end{bmatrix}$ be a standard generator for C where A is $k \times (n-k)$, and write $H = \begin{bmatrix} -A^T & I_{n-k} \end{bmatrix}$ for the parity-check matrix. Then:

1. *H is a generator of $C^\perp$.*
2. *$\dim(C^\perp) = n - k = \operatorname{rank} H$.*
3. *$C^{\perp\perp} = C$ and $\dim(C^\perp) + \dim C = n$.*

Proof. As in Theorem 8.8.6, let $D = \{\mathbf{v}H \mid \mathbf{v} \text{ in } F^{n-k}\}$ denote the code generated by H. Observe first that, for all $\mathbf{w}$ in F^n and all $\mathbf{u}$ in F^k, we have

$$\mathbf{w}\cdot(\mathbf{u}G) = \mathbf{w}(\mathbf{u}G)^T = \mathbf{w}(G^T\mathbf{u}^T) = (\mathbf{w}G^T)\cdot\mathbf{u}$$

Since $C = \{\mathbf{u}G \mid \mathbf{u} \text{ in } F^k\}$, this shows that $\mathbf{w}$ is in $C^\perp$ if and only if $(\mathbf{w}G^T)\cdot\mathbf{u} = 0$ for all $\mathbf{u}$ in F^k; if and only if[22] $\mathbf{w}G^T = \mathbf{0}$; if and only if $\mathbf{w}$ is in D (by Theorem 8.8.6). Thus $C^\perp = D$ and a similar argument shows that $D^\perp = C$.

1. H generates $C^\perp$ because $C^\perp = D = \{\mathbf{v}H \mid \mathbf{v} \text{ in } F^{n-k}\}$.
2. This follows from (1) because, as we observed above, $\operatorname{rank} H = n - k$.
3. Since $C^\perp = D$ and $D^\perp = C$, we have $C^{\perp\perp} = (C^\perp)^\perp = D^\perp = C$. Finally the second equation in (3) restates (2) because $\dim C = k$.

[21] For an elementary introduction, see V. Pless, *Introduction to the Theory of Error-Correcting Codes*, 3rd ed., (New York: Wiley, 1998).

[22] If $\mathbf{v}\cdot\mathbf{u} = 0$ for every $\mathbf{u}$ in F^k, then $\mathbf{v} = \mathbf{0}$—let $\mathbf{u}$ range over the standard basis of F^k.

□

We note in passing that, if C is a subspace of $\mathbb{R}^k$, we have $C + C^\perp = \mathbb{R}^k$ by the projection theorem (Theorem 8.1.3), and $C \cap C^\perp = \{\mathbf{0}\}$ because any vector $\mathbf{x}$ in $C \cap C^\perp$ satisfies $\|\mathbf{x}\|^2 = \mathbf{x} \cdot \mathbf{x} = 0$. However, this fails in general. For example, if $F = \mathbb{Z}_2$ and $C = \text{span}\{1010,\ 0101\}$ in F^4 then $C^\perp = C$, so $C + C^\perp = C = C \cap C^\perp$.

We conclude with one more example. If $F = \mathbb{Z}_2$, consider the standard matrix G below, and the corresponding parity-check matrix H:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The code $C = \{\mathbf{u}G \mid \mathbf{u} \text{ in } F^4\}$ generated by G has dimension $k = 4$, and is called the **Hamming (7, 4)-code**. The vectors in C are listed in the first table below. The dual code generated by H has dimension $n - k = 3$ and is listed in the second table.

C:

u	**u**G
0000	0000000
0001	0001011
0010	0010101
0011	0011110
0100	0100110
0101	0101101
0110	0110011
0111	0111000
1000	1000111
1001	1001100
1010	1010010
1011	1011001
1100	1100001
1101	1101010
1110	1110100
1111	1111111

$C^\perp$:

v	**v**H
000	0000000
001	1011001
010	1101010
011	0110011
100	1110100
101	0101101
110	0011110
111	1000111

Clearly each nonzero code word in C has weight at least 3, so C has minimum distance $d = 3$. Hence C can detect two errors and correct one error by Theorem 8.8.5. The dual code has minimum distance 4 and so can detect 3 errors and correct 1 error.

Exercises for 8.8

Exercise 8.8.1 Find all a in $\mathbb{Z}_{10}$ such that:

a. $a^2 = a$.

b. a has an inverse (and find the inverse).

c. $a^k = 0$ for some $k \geq 1$.

d. $a = 2^k$ for some $k \geq 1$.

e. $a = b^2$ for some b in $\mathbb{Z}_{10}$.

Exercise 8.8.2

a. Show that if $3a = 0$ in $\mathbb{Z}_{10}$, then necessarily $a = 0$ in $\mathbb{Z}_{10}$.

b. Show that $2a = 0$ in $\mathbb{Z}_{10}$ holds in $\mathbb{Z}_{10}$ if and only if $a = 0$ or $a = 5$.

Exercise 8.8.3 Find the inverse of:

a. 8 in $\mathbb{Z}_{13}$; b. 11 in $\mathbb{Z}_{19}$.

Exercise 8.8.4 If $ab = 0$ in a field F, show that either $a = 0$ or $b = 0$.

Exercise 8.8.5 Show that the entries of the last column of the multiplication table of $\mathbb{Z}_n$ are

$$0,\ n-1,\ n-2,\ \ldots,\ 2,\ 1$$

in that order.

Exercise 8.8.6 In each case show that the matrix A is invertible over the given field, and find A^{-1}.

a. $A = \begin{bmatrix} 1 & 4 \\ 2 & 1 \end{bmatrix}$ over $\mathbb{Z}_5$.

b. $A = \begin{bmatrix} 5 & 6 \\ 4 & 3 \end{bmatrix}$ over $\mathbb{Z}_7$.

Exercise 8.8.7 Consider the linear system $\begin{matrix} 3x + y + 4z = 3 \\ 4x + 3y + z = 1 \end{matrix}$. In each case solve the system by reducing the augmented matrix to reduced row-echelon form over the given field:

a. $\mathbb{Z}_5$ b. $\mathbb{Z}_7$

Exercise 8.8.8 Let K be a vector space over $\mathbb{Z}_2$ with basis $\{1, t\}$, so $K = \{a + bt \mid a, b, \text{ in } \mathbb{Z}_2\}$. It is known that K becomes a field of four elements if we define $t^2 = 1 + t$. Write down the multiplication table of K.

Exercise 8.8.9 Let K be a vector space over $\mathbb{Z}_3$ with basis $\{1, t\}$, so $K = \{a + bt \mid a, b, \text{ in } \mathbb{Z}_3\}$. It is known that K becomes a field of nine elements if we define $t^2 = -1$ in $\mathbb{Z}_3$. In each case find the inverse of the element x of K:

a. $x = 1 + 2t$ b. $x = 1 + t$

Exercise 8.8.10 How many errors can be detected or corrected by each of the following binary linear codes?

a. $C = \{0000000,\ 0011110,\ 0100111,\ 0111001,\ 1001011,\ 1010101,\ 1101100,\ 1110010\}$

b. $C = \{0000000000,\ 0010011111,\ 0101100111,\ 0111111000,\ 1001110001,\ 1011101110,\ 1100010110,\ 1110001001\}$

Exercise 8.8.11

a. If a binary linear $(n,\ 2)$-code corrects one error, show that $n \geq 5$. [*Hint*: Hamming bound.]

b. Find a $(5,\ 2)$-code that corrects one error.

Exercise 8.8.12

a. If a binary linear $(n,\ 3)$-code corrects two errors, show that $n \geq 9$. [*Hint*: Hamming bound.]

b. If $G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$, show that the binary $(10,\ 3)$-code generated by G corrects two errors. [It can be shown that no binary $(9,\ 3)$-code corrects two errors.]

Exercise 8.8.13

a. Show that no binary linear $(4,\ 2)$-code can correct single errors.

b. Find a binary linear $(5,\ 2)$-code that can correct one error.

Exercise 8.8.14 Find the standard generator matrix G and the parity-check matrix H for each of the following systematic codes:

a. $\{00000,\ 11111\}$ over $\mathbb{Z}_2$.

b. Any systematic $(n,\ 1)$-code where $n \geq 2$.

c. The code in Exercise 8.8.10(a).

d. The code in Exercise 8.8.10(b).

Exercise 8.8.15 Let $\mathbf{c}$ be a word in F^n. Show that $B_t(\mathbf{c}) = \mathbf{c} + B_t(\mathbf{0})$, where we write

$$\mathbf{c} + B_t(\mathbf{0}) = \{\mathbf{c} + \mathbf{v} \mid \mathbf{v} \text{ in } B_t(\mathbf{0})\}$$

Exercise 8.8.16 If a $(n,\ k)$-code has two standard generator matrices G and G_1, show that $G = G_1$.

Exercise 8.8.17 Let C be a binary linear n-code (over $\mathbb{Z}_2$). Show that either each word in C has even weight, or half the words in C have even weight and half have odd weight. [*Hint*: The dimension theorem.]

8.9 An Application to Quadratic Forms

An expression like $x_1^2 + x_2^2 + x_3^2 - 2x_1x_3 + x_2x_3$ is called a quadratic form in the variables x_1, x_2, and x_3. In this section we show that new variables y_1, y_2, and y_3 can always be found so that the quadratic form, when expressed in terms of the new variables, has no cross terms y_1y_2, y_1y_3, or y_2y_3. Moreover, we do this for forms involving any finite number of variables using orthogonal diagonalization. This has far-reaching applications; quadratic forms arise in such diverse areas as statistics, physics, the theory of functions of several variables, number theory, and geometry.

Definition 8.21 Quadratic Form

*A **quadratic form** q in the n variables $x_1,\ x_2,\ \ldots,\ x_n$ is a linear combination of terms $x_1^2,\ x_2^2,\ \ldots,\ x_n^2$, and cross terms $x_1x_2,\ x_1x_3,\ x_2x_3,\ \ldots$.*

If $n = 3$, q has the form

$$q = a_{11}x_1^2 + a_{22}x_2^2 + a_{33}x_3^2 + a_{12}x_1x_2 + a_{21}x_2x_1 + a_{13}x_1x_3 + a_{31}x_3x_1 + a_{23}x_2x_3 + a_{32}x_3x_2$$

In general

$$q = a_{11}x_1^2 + a_{22}x_2^2 + \cdots + a_{nn}x_n^2 + a_{12}x_1x_2 + a_{13}x_1x_3 + \cdots$$

This sum can be written compactly as a matrix product

$$q = q(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$$

where $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_n)$ is thought of as a column, and $A = \left[a_{ij}\right]$ is a real $n \times n$ matrix. Note that if $i \neq j$, two separate terms $a_{ij}x_ix_j$ and $a_{ji}x_jx_i$ are listed, each of which involves x_ix_j, and they can (rather cleverly) be replaced by

$$\tfrac{1}{2}(a_{ij} + a_{ji})x_ix_j \quad \text{and} \quad \tfrac{1}{2}(a_{ij} + a_{ji})x_jx_i$$

respectively, *without altering the quadratic form*. Hence there is no loss of generality in assuming that x_ix_j and x_jx_i have the same coefficient in the sum for q. In other words, **we may assume that A is symmetric**.