Chapter 9

Algebraic Coding Theory

In this chapter we will discuss some applications of techniques from computational algebra and algebraic geometry to problems in coding theory. After a preliminary section on the arithmetic of finite fields, we will introduce some basic terminology for describing error-correcting codes. We will study two important classes of examples—linear codes and cyclic codes—where the set of codewords possesses additional algebraic structure, and we will use this structure to develop good encoding and decoding algorithms.

§1 Finite Fields

To make our presentation as self-contained as possible, in this section we will develop some of the basic facts about the arithmetic of finite fields. We will do this almost "from scratch," without using the general theory of field extensions. However, we will need to use some elementary facts about finite groups and quotient rings. Readers who have seen this material before may wish to proceed directly to §2. More complete treatments of this classical subject can be found in many texts on abstract algebra or Galois theory.

The most basic examples of finite fields are the *prime fields* $\mathbb{F}_p = \mathbb{Z}/\langle p \rangle$, where p is any prime number, but there are other examples as well. To construct them, we will need to use the following elementary fact.

Exercise 1. Let k be any field, and let $g \in k[x]$ be an *irreducible* polynomial (that is, a non-constant polynomial which is not the product of two nonconstant polynomials in $k[x]$). Show that the ideal $\langle g \rangle \subset k[x]$ is a maximal ideal. Deduce that $k[x]/\langle g \rangle$ is a field if g is irreducible.

For example, let $p = 3$ and consider the polynomial $g = x^2 + x + 2 \in \mathbb{F}_3[x]$. Since g is a quadratic polynomial with no roots in $\mathbb{F}_3$, g is irreducible in $\mathbb{F}_3[x]$. By Exercise 1, the ideal $\langle g \rangle$ is maximal, hence $\mathbb{F} = \mathbb{F}_3[x]/\langle g \rangle$ is a field. As we discussed in Chapter 2, the elements of a quotient ring such as $\mathbb{F}$ are in one-to-one correspondence with the possible remainders on division

by g. Hence the elements of $\mathbb{F}$ are the cosets of the polynomials $ax + b$, where a, b are arbitrary in $\mathbb{F}_3$. As a result, $\mathbb{F}$ is a field of $3^2 = 9$ elements.

To distinguish more clearly between polynomials and the elements of our field, we will write α for the element of $\mathbb{F}$ represented by the polynomial x. Thus every element of $\mathbb{F}$ has the form $a\alpha + b$ for $a, b \in \mathbb{F}_3$. Also, note that α satisfies the equation $g(\alpha) = \alpha^2 + \alpha + 2 = 0$.

The addition operation in $\mathbb{F}$ is the obvious one: $(a\alpha + b) + (a'\alpha + b') = (a + a')\alpha + (b + b')$. As in Chapter 2 §2, we can compute products in $\mathbb{F}$ by multiplication of polynomials in α, subject to the relation $g(\alpha) = 0$. For instance, you should verify that in $\mathbb{F}$

$$(\alpha + 1) \cdot (2\alpha + 1) = 2\alpha^2 + 1 = \alpha$$

(recall that the coefficients of these polynomials are elements of the field $\mathbb{F}_3$, so that $1 + 2 = 0$). Using this method, we may compute all the powers of α in $\mathbb{F}$, and we find

$$\begin{array}{llllllll} \alpha^2 & = & 2\alpha + 1 & & \alpha^3 & = & 2\alpha + 2 \\ \alpha^4 & = & 2 & & \alpha^5 & = & 2\alpha \\ \alpha^6 & = & \alpha + 2 & & \alpha^7 & = & \alpha + 1, \end{array} \tag{1.1}$$

and $\alpha^8 = 1$. For future reference, we note that this computation also shows that the multiplicative group of nonzero elements of $\mathbb{F}$ is a cyclic group of order 8, generated by α.

The construction of $\mathbb{F}$ in this example may be generalized in the following way. Consider the polynomial ring $\mathbb{F}_p[x]$, and let $g \in \mathbb{F}_p[x]$ be an irreducible polynomial of degree n. The ideal $\langle g \rangle$ is maximal by Exercise 1, so the quotient ring $\mathbb{F} = \mathbb{F}_p[x]/\langle g \rangle$ is a field. The elements of $\mathbb{F}$ may be represented by the cosets modulo $\langle g \rangle$ of the polynomials of degree $n - 1$ or less: $a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, a_i \in \mathbb{F}_p$. Since the a_i are arbitrary, this implies that $\mathbb{F}$ contains p^n distinct elements.

Exercise 2.

a. Show that $g = x^4 + x + 1$ is irreducible in $\mathbb{F}_2[x]$. How many elements are there in the field $\mathbb{F} = \mathbb{F}_2[x]/\langle g \rangle$?
b. Writing α for the element of $\mathbb{F}$ represented by x as above, compute all the distinct powers of α.
c. Show that $\mathbb{K} = \{0, 1, \alpha^5, \alpha^{10}\}$ is a field with four elements contained in $\mathbb{F}$.
d. Is there a field with exactly eight elements contained in $\mathbb{F}$? Are there any other subfields? (For the general pattern, see Exercise 10 below.)

In general we may ask what the possible sizes (numbers of elements) of finite fields are. The following proposition gives a necessary condition.

(1.2) Proposition. *Let $\mathbb{F}$ be a finite field. Then $|\mathbb{F}| = p^n$ where p is some prime number and $n \geq 1$.*

PROOF. Since $\mathbb{F}$ is a field, it contains a multiplicative identity, which we will denote by 1 as usual. Since $\mathbb{F}$ is finite, 1 must have finite additive order: say p is the smallest positive integer such that $p \cdot 1 = 1 + \cdots + 1 = 0$ (p summands). The integer p must be prime. (Otherwise, if $p = mn$ with $m, n > 1$, then we would have $p \cdot 1 = (m \cdot 1)(n \cdot 1) = 0$ in $\mathbb{F}$. But since $\mathbb{F}$ is a field, this would imply $m \cdot 1 = 0$, or $n \cdot 1 = 0$, which is not possible by the minimality of p.) We leave it to the reader to check that the set of elements of the form $m \cdot 1$, $m = 0, 1, \ldots, p-1$ in $\mathbb{F}$ is a subfield $\mathbb{K}$ isomorphic to $\mathbb{F}_p$. See Exercise 9 below.

The axioms for fields imply that if we consider the addition operation on $\mathbb{F}$ together with scalar multiplication of elements of $\mathbb{F}$ by elements from $\mathbb{K} \subset \mathbb{F}$, then $\mathbb{F}$ has the structure of a vector space over $\mathbb{K}$. Since $\mathbb{F}$ is a finite set, it must be finite-dimensional as a vector space over $\mathbb{K}$. Let n be its dimension (the number of elements in any basis), and let $\{a_1, \ldots, a_n\} \subset \mathbb{F}$ be any basis. Every element of $\mathbb{F}$ can be expressed in exactly one way as a linear combination $c_1 a_1 + \cdots + c_n a_n$, where $c_1, \ldots, c_n \in \mathbb{K}$. There are p^n such linear combinations, which concludes the proof. □

To construct finite fields, we will always consider quotient rings $\mathbb{F}_p[x]/\langle g \rangle$ where g is an irreducible polynomial in $\mathbb{F}_p[x]$. There is no loss of generality in doing this—every finite field can be obtained this way. See Exercise 11 below.

We will show next that for each prime p and each $n \geq 1$, there exist finite fields of every size p^n by counting the irreducible polynomials of fixed degree in $\mathbb{F}_p[x]$. First note that it is enough to consider monic polynomials, since we can always multiply by a constant in $\mathbb{F}_p$ to make the leading coefficient of a polynomial equal 1. There are exactly p^n distinct monic polynomials $x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ of degree n in $\mathbb{F}_p[x]$. Consider the *generating function* for this enumeration by degree: the power series in u in which the coefficient of u^n equals the number of monic polynomials of degree n, namely p^n. This is the left hand side in (1.3) below. We treat this as a purely formal series and disregard questions of convergence. The formal geometric series summation formula yields

$$\sum_{n=0}^{\infty} p^n u^n = \frac{1}{1-pu}. \tag{1.3}$$

Each monic polynomial factors *uniquely* in $\mathbb{F}_p[x]$ into a product of monic irreducibles. For each n, let N_n be the number of monic irreducibles of degree n in $\mathbb{F}_p[x]$. In factorizations of the form $g = g_1 \cdot g_2 \cdots g_m$ where the g_i are irreducible (but not necessarily distinct) of degrees n_i, we have N_{n_i} choices for the factor g_i for each i. The total degree of g is the sum of the degrees of the factors.

Exercise 3. By counting factorizations as above, show that the number of monic polynomials of degree n (i.e. p^n) can also be expressed as the coefficient of u^n in the formal infinite product

$$(1 + u + u^2 + \cdots)^{N_1} \cdot (1 + u^2 + u^4 + \cdots)^{N_2} \cdots = \prod_{k=1}^{\infty} \frac{1}{(1 - u^k)^{N_k}},$$

where the equality between the left- and right-hand sides comes from the formal geometric series summation formula. Hint: The term in the product with index k accounts for factors of degree k in polynomials.

Hence combining (1.3) with the result of Exercise 3, we obtain the generating function identity

$$\prod_{k=1}^{\infty} \frac{1}{(1 - u^k)^{N_k}} = \frac{1}{1 - pu}. \tag{1.4}$$

(1.5) Proposition. *We have $p^n = \sum_{k|n} kN_k$, where the sum extends over all positive divisors k of the integer n.*

Proof. Formally taking logarithmic derivatives and multiplying the results by u, from (1.4) we arrive at the identity $\sum_{k=1}^{\infty} kN_k u^k/(1 - u^k) = pu/(1 - pu)$. Using formal geometric series again, this equality can be rewritten as

$$\sum_{k=1}^{\infty} kN_k(u^k + u^{2k} + \cdots) = pu + p^2u^2 + \cdots .$$

The proposition follows by comparing the coefficients of u^n on both sides of this last equation. □

Exercise 4. (For readers with some background in elementary number theory.) Use Proposition (1.5) and the Möbius inversion formula to derive a general formula for N_n.

We will show that $N_n > 0$ for all $n \geq 1$. For $n = 1$, we have $N_1 = p$ since all $x - \beta$, $\beta \in \mathbb{F}_p$ are irreducible. Then Proposition (1.5) implies that $N_2 = (p^2 - p)/2 > 0$, $N_3 = (p^3 - p)/3 > 0$, and $N_4 = (p^4 - p^2)/4 > 0$.

Arguing by contradiction, suppose that $N_n = 0$ for some n. We may assume $n \geq 5$ by the above. Then from Proposition (1.5),

$$p^n = \sum_{k|n, 0<k<n} kN_k. \tag{1.6}$$

We can estimate the size of the right-hand side and derive a contradiction from (1.6) as follows. We write $\lfloor A \rfloor$ for the greatest integer less than or equal to A. Since $N_k \leq p^k$ for all k (the irreducibles are a subset of the

whole collection monic polynomials), and any positive proper divisor k of n is at most $\lfloor n/2 \rfloor$, we have

$$p^n \leq \lfloor n/2 \rfloor \sum_{k=0}^{\lfloor n/2 \rfloor} p^k.$$

Applying the finite geometric sum formula, the right-hand side equals

$$\lfloor n/2 \rfloor (p^{\lfloor n/2 \rfloor + 1} - 1)/(p-1) \leq \lfloor n/2 \rfloor p^{\lfloor n/2 \rfloor + 1}.$$

Hence

$$p^n \leq \lfloor n/2 \rfloor p^{\lfloor n/2 \rfloor + 1}.$$

Dividing each side by $p^{\lfloor n/2 \rfloor}$, we obtain

$$p^{n - \lfloor n/2 \rfloor} \leq \lfloor n/2 \rfloor p.$$

But this is clearly false for all p and all $n \geq 5$. Hence $N_n > 0$ for all n, and as a result we have the following fact.

(1.7) Theorem. *For all primes p and all $n \geq 1$, there exist finite fields $\mathbb{F}$ with $|\mathbb{F}| = p^n$.*

From the examples we have seen and from the proof of Theorem (1.7), it might appear that there are several different finite fields of a given size, since there will usually be more than one irreducible polynomial g of a given degree in $\mathbb{F}_p[x]$ to use in constructing quotients $\mathbb{F}_p[x]/\langle g \rangle$. But consider the following example.

Exercise 5. By Proposition (1.5), there are $(2^3 - 2)/3 = 2$ monic irreducible polynomials of degree 3 in $\mathbb{F}_2[x]$, namely $g_1 = x^3 + x + 1$, and $g_2 = x^3 + x^2 + 1$. Hence $\mathbb{K}_1 = \mathbb{F}_2[x]/\langle g_1 \rangle$ and $\mathbb{K}_2 = \mathbb{F}_2[x]/\langle g_2 \rangle$ are two finite fields with 8 elements. We claim, however, that these fields are *isomorphic*.

a. Writing α for the coset of x in $\mathbb{K}_1$ (so $g_1(\alpha) = 0$ in $\mathbb{K}_1$), show that $g_2(\alpha + 1) = 0$ in $\mathbb{K}_1$.
b. Use this observation to derive an isomorphism between $\mathbb{K}_1$ and $\mathbb{K}_2$ (that is, a one-to-one, onto mapping that preserves sums and products).

The general pattern is the same.

(1.8) Theorem. *Let $\mathbb{K}_1$ and $\mathbb{K}_2$ be two fields with $|\mathbb{K}_1| = |\mathbb{K}_2| = p^n$. Then $\mathbb{K}_1$ and $\mathbb{K}_2$ are isomorphic.*

See Exercise 12 below for one way to prove this. Because of (1.8), it makes sense to adopt the uniform notation $\mathbb{F}_{p^n}$ for any field of order p^n, and we will use this convention for the remainder of the chapter. When we do computations in $\mathbb{F}_{p^n}$, however, we will always use an explicit monic irreducible polynomial $g(x)$ of degree n as in the examples above.

The next general fact we will consider is also visible in (1.1) and in the other examples we have encountered.

(1.9) Theorem. *Let $\mathbb{F} = \mathbb{F}_{p^n}$ be a finite field. The multiplicative group of nonzero elements of $\mathbb{F}$ is cyclic of order $p^n - 1$.*

PROOF. The statement about the order of the multiplicative group is clear since we are omitting the single element 0. Write $m = p^n - 1$. By Lagrange's Theorem for finite groups ([Her]), every element $\beta \in \mathbb{F} \setminus \{0\}$ is a root of the polynomial equation $x^m = 1$, and the multiplicative order of each is a divisor of m. We must show there is some element of order exactly m to conclude the proof. Consider the prime factorization of m, say $m = q_1^{e_1} \cdots q_k^{e_k}$. Let $m_i = m/q_i$. Since the polynomial $x^{m_i} - 1$ has at most m_i roots in the field $\mathbb{F}$, there is some $\beta_i \in \mathbb{F}$ such that $\beta_i^{m_i} \neq 1$. In Exercise 6 below, you will show that $\gamma_i = \beta_i^{m/q_i^{e_i}}$ has multiplicative order exactly $q_i^{e_i}$ in $\mathbb{F}$. It follows that the product $\gamma_1\gamma_2 \cdots \gamma_k$ has order m, since the $q_i^{e_i}$ are relatively prime. □

Exercise 6. In this exercise you will supply details for the final two claims in the proof of Theorem (1.9).

a. Using the notation from the proof, show that $\gamma_i = \beta_i^{m/q_i^{e_i}}$ has multiplicative order exactly $q_i^{e_i}$ in $\mathbb{F}$. (That is, show that $\gamma_i^{q_i^{e_i}} = 1$, but that $\gamma_i^k \neq 1$ for all $k = 1, \ldots, q_i^{e_i} - 1$.)
b. Let γ_1, γ_2 be elements of a finite abelian group. Suppose that the orders of γ_1 and γ_2 (n_1 and n_2 respectively) are relatively prime. Show that the order of the product $\gamma_1\gamma_2$ is equal to n_1n_2.

A generator for the multiplicative group of $\mathbb{F}_{p^n}$ is called a *primitive element*. In the fields studied in (1.1) and in Exercise 2, the polynomials g were chosen so that their roots were primitive elements of the corresponding finite field. This will not be true for *all* choices of irreducible g of a given degree n in $\mathbb{F}_p[x]$.

Exercise 7. For instance, consider the polynomial $g = x^2 + 1$ in $\mathbb{F}_3[x]$. Check that g is irreducible, so that $\mathbb{K} = \mathbb{F}_3[x]/\langle g \rangle$ is a field with 9 elements. However the coset of x in $\mathbb{K}$ is not a primitive element. (Why not? what is its multiplicative order?)

For future reference, we also include the following fact about finite fields.

Exercise 8. Suppose that $\beta \in \mathbb{F}_{p^n}$ is neither 0 nor 1. Then show that $\sum_{j=0}^{p^n-2} \beta^j = 0$. Hint: What is $(x^{p^n-1} - 1)/(x - 1)$?

To conclude this section, we indicate one direct method for performing finite field arithmetic in Maple. Maple provides a built-in facility (via the

`mod` operator) by which polynomial division, row operations on matrices, resultant computations, etc. can be done using coefficients in finite fields. When we construct a quotient ring $\mathbb{F}_p[x]/\langle g\rangle$ the coset of x becomes a root of the equation $g = 0$ in the quotient. In Maple, the elements of a finite field can be represented as (cosets of) polynomials in `RootOf` expressions (see Chapter 2, §1). For example, to declare the field $\mathbb{F}_8 = \mathbb{F}_2[x]/\langle x^3 + x + 1\rangle$, we could use

```
alias(alpha = RootOf(x^3 + x + 1));
```

Then polynomials in `alpha` represent elements of the field $\mathbb{F}_8$ as before. Arithmetic in the finite field can be performed as follows. For instance, suppose we want to compute $b^3 + b$, where $b = \alpha + 1$. Entering

```
b := alpha + 1;
Normal(b^3 + b) mod 2;
```

yields

$$alpha^2 + alpha + 1.$$

The `Normal` function computes the normal form for the element of the finite field by expanding out $b^3 + b$ as a polynomial in α, then finding the remainder on division by $\alpha^3 + \alpha + 1$, using coefficients in $\mathbb{F}_2$.

A technical point: You may have noticed that the `Normal` function name here is *capitalized*. There is also an uncapitalized `normal` function in Maple which can be used for algebraic simplification of expressions. We *do not* want that function here, however, because we want the function call to be passed *unevaluated* to `mod`, and all the arithmetic to be performed within the mod environment. Maple uses capitalized names consistently for unevaluated function calls in this situation. Using the command `normal(b^3 + b) mod 2` would instruct Maple to simplify $b^3 + b$, *then reduce mod 2*, which does not yield the correct result in this case. Try it!

Additional Exercises for §1

Exercise 9. Verify the claim made in the proof of Proposition (1.2) that if $\mathbb{F}$ is a field with p^n elements, then $\mathbb{F}$ has a subfield

$$\mathbb{K} = \{0, 1, 2 \cdot 1, \ldots, (p-1) \cdot 1\}$$

isomorphic to $\mathbb{F}_p$.

Exercise 10. Using Theorem (1.9), show that $\mathbb{F}_{p^n}$ contains a subfield $\mathbb{F}_{p^m}$ if and only if m is a divisor of n. Hint: By (1.9), the multiplicative group of the subfield is a subgroup of the multiplicative cyclic group $\mathbb{F}_{p^m} \setminus \{0\}$. What are the orders of subgroups of a cyclic group of order $p^m - 1$?

Exercise 11. In this exercise, you will show that every finite field $\mathbb{F}$ can be obtained (up to isomorphism) as a quotient $\mathbb{F} \cong \mathbb{F}_p[x]/\langle g \rangle$ for some irreducible $g \in \mathbb{F}_p[x]$. For this exercise, we will need the fundamental theorem of ring homomorphisms (see e.g., [CLO] Chapter 5, §2, Exercise 16). Let $\mathbb{F}$ be a finite field, and say $|\mathbb{F}| = p^n$ (see Proposition (1.2)). Let α be a primitive element for $\mathbb{F}$ (see (1.9)). Consider the ring homomorphism defined by

$$\begin{aligned} \varphi : \mathbb{F}_p[x] &\to \mathbb{F} \\ x &\mapsto \alpha. \end{aligned}$$

a. Explain why φ must be onto.
b. Deduce that the kernel of φ must have the form $\ker(\varphi) = \langle g \rangle$ for some irreducible monic polynomial $g \in k[x]$. (The monic generator is called the *minimal polynomial* of α over $\mathbb{F}_p$.)
c. Apply the fundamental theorem to show that

$$\mathbb{F} \cong \mathbb{F}_p[x]/\langle g \rangle.$$

Exercise 12. In this exercise, you will develop one proof of Theorem (1.8), using Theorem (1.9) and the previous exercise. Let $\mathbb{K}$ and $\mathbb{L}$ be two fields with p^n elements. Let β be a primitive element for $\mathbb{L}$, and let $g \in \mathbb{F}_p[x]$ be the minimal polynomial of β over $\mathbb{F}_p$, so that $\mathbb{L} \cong \mathbb{F}_p[x]/\langle g \rangle$ (Exercise 11).
a. Show that g must divide the polynomial $x^{p^n} - x$ in $\mathbb{F}_p[x]$. (Use (1.9).)
b. Show that $x^{p^n} - x$ splits completely into linear factors in $\mathbb{K}[x]$:

$$x^{p^n} - x = \prod_{\alpha \in \mathbb{K}} (x - \alpha).$$

c. Deduce that there is some $\alpha \in \mathbb{K}$ which is a root of $g = 0$.
d. From part c, deduce that $\mathbb{K}$ is *also* isomorphic to $\mathbb{F}_p[x]/\langle g \rangle$. Hence, $\mathbb{K} \cong \mathbb{L}$.

Exercise 13. Find irreducible polynomials g in the appropriate $\mathbb{F}_p[x]$, such that $\mathbb{F}_p[x]/\langle g \rangle \cong \mathbb{F}_{p^n}$, and such that $\alpha = [x]$ is a primitive element in $\mathbb{F}_{p^n}$ for each $p^n \leq 64$. (Note: The cases $p^n = 8, 9, 16$ are considered in the text. Extensive tables of such polynomials have been constructed for use in coding theory. See for instance [PH].)

Exercise 14. (The Frobenius Automorphism) Let $\mathbb{F}_q$ be a finite field. By Exercise 10, $\mathbb{F}_q \subset \mathbb{F}_{q^m}$ for each $m \geq 1$. Consider the mapping $F : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ defined by $F(x) = x^q$.
a. Show that F is one-to-one and onto, and that $F(x + y) = F(x) + F(y)$ and $F(xy) = F(x)F(y)$ for all $x, y \in \mathbb{F}_{q^m}$. (In other words, F is an *automorphism* of the field $\mathbb{F}_{q^m}$.)
b. Show that $F(x) = x$ if and only if $x \in \mathbb{F}_q \subset \mathbb{F}_{q^m}$.

For readers familiar with Galois theory, we mention that the Frobenius automorphism F generates the Galois group of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$—a cyclic group of order m.

§2 Error-Correcting Codes

In this section, we will introduce some of the basic standard notions from algebraic coding theory. For more complete treatments of the subject, we refer the reader to [vLi], [Bla], or [MS].

Communication of information often takes place over "noisy" channels which can introduce errors in the transmitted message. This is the case for instance in satellite transmissions, in the transfer of information within computer systems, and in the process of storing information (numeric data, music, images, etc.) on tape, on compact disks or other media, and retrieving it for use at a later time. In these situations, it is desirable to *encode* the information in such a way that errors can be detected and/or corrected when they occur. The design of coding schemes, together with efficient techniques for encoding and decoding (i.e. recovering the original message from its encoded form) is one of the main goals of coding theory.

In some situations, it might also be desirable to encode information in such a way that unauthorized readers of the received message will not be able to decode it. The construction of codes for secrecy is the domain of *cryptography*, a related but distinct field that will *not* be considered here. Interestingly enough, ideas from number theory and algebraic geometry have assumed a major role there as well. The book [Kob] includes some applications of computational algebraic geometry in modern cryptography.

In this chapter, we will study one specific type of code, in which all information to be encoded consists of strings or *words* of some fixed length k using symbols from a fixed alphabet, and all encoded messages are divided into strings called *codewords* of a fixed block length n, using symbols from the same alphabet. In order to detect and/or correct errors, some *redundancy* must be introduced in the encoding process, hence we will always have $n > k$.

Because of the design of most electronic circuitry, it is natural to consider a binary alphabet consisting of the two symbols $\{0, 1\}$, and to identify the alphabet with the finite field $\mathbb{F}_2$. As in §1, strings of r bits (thought of as the coefficients in a polynomial of degree $r-1$) can also represent elements of a field $\mathbb{F}_{2^r}$, and it will be advantageous in some cases to think of $\mathbb{F}_{2^r}$ as the alphabet. But the constructions we will present are valid with an arbitrary finite field $\mathbb{F}_q$ as the alphabet.

In mathematical terms, the encoding process for a string from the message will be a one-to-one function $E : \mathbb{F}_q^k \to \mathbb{F}_q^n$. The image $C = E(\mathbb{F}_q^k) \subset \mathbb{F}_q^n$ is referred to as the set of codewords, or more succinctly as *the code*. Mathematically, the decoding operation can be viewed as a

function $D : \mathbb{F}_q^n \to \mathbb{F}_q^k$ such that $D \circ E$ is the identity on $\mathbb{F}_q^k$. (This is actually an over-simplification—most real-world decoding functions will also return something like an "error" value in certain situations.)

In principle, the set of codewords can be an arbitrary subset of $\mathbb{F}_q^n$. However, we will almost always restrict our attention to a class of codes with additional structure that is very convenient for encoding and decoding. This is the class of *linear codes*. By definition, a linear code is one where the set of codewords C forms a vector subspace of $\mathbb{F}_q^n$ of dimension k. In this case, as encoding function $E : \mathbb{F}_q^k \to \mathbb{F}_q^n$ we may use a linear mapping whose image is the subspace C. The matrix of E with respect to the standard bases in the domain and target is often called the *generator matrix* G corresponding to E.

It is customary in coding theory to write generator matrices for linear codes as $k \times n$ matrices and to view the strings in $\mathbb{F}_q^k$ as *row vectors* w. Then the encoding operation can be viewed as matrix multiplication of a row vector on the right by the generator matrix, and the rows of G form a basis for C. As always in linear algebra, the subspace C may also be described as the set of solutions of a system of $n - k$ independent linear equations in n variables. The matrix of coefficients of such a system is often called a *parity check matrix* for C. This name comes from the fact that one simple error-detection scheme for binary codes is to require that all codewords have an even (or odd) number of nonzero digits. If one bit error (in fact, any odd number of errors) is introduced in transmission, that fact can be recognized by multiplication of the received word by the parity check matrix $H = (\,1 \quad 1 \quad \cdots \quad 1\,)^T$. The parity check matrix for a linear code can be seen as an extension of this idea, in which more sophisticated tests for the validity of the received word are performed by multiplication by the parity check matrix.

Exercise 1. Consider the linear code C with $n = 4$, $k = 2$ given by the generator matrix

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

a. Show that since we have only the two scalars $0, 1 \in \mathbb{F}_2$ to use in making linear combinations, there are exactly four elements of C:

$$(0,0)G = (0,0,0,0), \qquad (1,0)G = (1,1,1,1),$$
$$(0,1)G = (1,0,1,0), \qquad (1,1)G = (0,1,0,1).$$

b. Show that

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

is a parity check matrix for C by verifying that $xH = 0$ for all $x \in C$.

Exercise 2. Let $\mathbb{F}_4 = \mathbb{F}_2[\alpha]/\langle \alpha^2 + \alpha + 1 \rangle$, and consider the linear code C in $\mathbb{F}_4^5$ with generator matrix

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

How many distinct codewords are there in C? Find them. Also find a parity check matrix for C. Hint: Recall from linear algebra that there is a general procedure using matrix operations for finding a system of linear equations defining a given subspace.

To study the error-correcting capability of codes, we need a measure of how close elements of $\mathbb{F}_q^n$ are, and for this we will use the *Hamming distance*. Let $x, y \in \mathbb{F}_q^n$. Then the Hamming distance between x and y is defined to be

$$d(x, y) = |\{i, 1 \le i \le n : x_i \ne y_i\}|.$$

For instance, if $x = (0, 0, 1, 1, 0)$, and $y = (1, 0, 1, 0, 0)$ in $\mathbb{F}_2^5$, then $d(x, y) = 2$ since only the first and fourth bits in x and y differ.

Let 0 denote the zero vector in $\mathbb{F}_q^n$ and let $x \in \mathbb{F}_q^n$ be arbitrary. Then $d(x, 0)$, the number of nonzero components in x, is called the *weight* of x and denoted $\mathrm{wt}(x)$.

Exercise 3. Verify that the Hamming distance has all the properties of a *metric* or *distance function* on $\mathbb{F}_q^n$. (That is, show $d(x, y) \ge 0$ for all x, y and $d(x, y) = 0$ if and only if $x = y$, the symmetry property $d(x, y) = d(y, x)$ holds for all x, y, and the triangle inequality $d(x, y) \le d(x, z) + d(z, y)$ is valid for all x, y, z.)

Given $x \in \mathbb{F}_q^n$, we will denote by $B_r(x)$ the closed ball of radius r (in the Hamming distance) centered at x:

$$B_r(x) = \{y \in \mathbb{F}_q^n : d(y, x) \le r\}.$$

(In other words, $B_r(x)$ is the set of y differing from x in at most r components.)

The Hamming distance gives a simple but extremely useful way to measure the error-correcting capability of a code. Namely, suppose that every pair of distinct codewords x, y in a code $C \subset \mathbb{F}_q^n$ satisfies $d(x, y) \ge d$ for some $d \ge 1$. If a codeword x is transmitted and errors are introduced, we can view the received word as $z = x + e$ for some nonzero error vector e. If $\mathrm{wt}(e) = d(x, z) \le d - 1$, then under our hypothesis z is not another codeword. Hence any error vector of weight at most $d - 1$ can be *detected*.

Moreover if $d \ge 2t + 1$ for some $t \ge 1$, then for any $z \in \mathbb{F}_q^n$, by the triangle inequality, $d(x, z) + d(z, y) \ge d(x, y) \ge 2t + 1$. It follows immediately that either $d(x, z) > t$ or $d(y, z) > t$, so $B_t(x) \cap B_t(y) = \emptyset$. As a result the only codeword in $B_t(x)$ is x itself. In other words, if an error vector of weight at most t is introduced in transmission of a codeword, those

errors can be *corrected* by the *nearest neighbor decoding function*

$$D(x) = E^{-1}(c), \text{ where } c \in C \text{ minimizes } d(x, c)$$

(and an "error" value if there is no unique closest element in C).

From this discussion it is clear that the *minimum distance*

$$d = \min\{d(x, y) : x \neq y \in C\}$$

is an important parameter of codes, and our observations above can be summarized in the following way.

(2.1) Proposition. *Let C be a code with minimum distance d. All error vectors of weight $\leq d - 1$ can be detected. Moreover, if $d \geq 2t + 1$, then all error vectors of weight $\leq t$ can be corrected by nearest neighbor decoding.*

Since the minimum distance of a code contains so much information, it is convenient that for linear codes we need only examine the codewords themselves to determine this parameter.

Exercise 4. Show that for any linear code C the minimum distance d is the same as $\min_{x \in C \setminus \{0\}} |\{i : x_i \neq 0\}|$ (the minimum number of nonzero entries in a nonzero codeword). Hint: Since the set of codewords is closed under vector sums, $x - y \in C$ whenever x and y are.

The *Hamming codes* form a famous family of examples with interesting error-correcting capabilities. One code in the family is a code over $\mathbb{F}_2$ with $n = 7$, $k = 4$. (The others are considered in Exercise 11 below.) For this Hamming code, the generator matrix is

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{2.2}$$

For example $w = (1, 1, 0, 1) \in \mathbb{F}_2^4$ is encoded by multiplication on the right by G, yielding $E(w) = wG = (1, 1, 0, 1, 0, 0, 1)$. From the form of the first four columns of G, the first four components of $E(w)$ will always consist of the four components of w itself.

The reader should check that the 7×3 matrix

$$H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

has rank 3 and satisfies $GH = 0$. Hence H is a parity check matrix for the Hamming code (why?). It is easy to check directly that each of the 15 nonzero codewords of the Hamming code contains at least 3 nonzero components. This implies that $d(x, y)$ is at least 3 when $x \neq y$. Hence the minimum distance of the Hamming code is $d = 3$, since there are exactly three nonzero entries in row 1 of G for example. By Proposition (2.1), any error vector of weight 2 or less can be detected, and any error vector of weight 1 can be corrected by nearest neighbor decoding. The following exercise gives another interesting property of the Hamming code.

Exercise 5. Show that the balls of radius 1 centered at each of the words of the Hamming code are pairwise disjoint, and cover $\mathbb{F}_2^7$ completely. (A code C with minimum distance $d = 2t + 1$ is called a *perfect* code if the union of the balls of radius t centered at the codewords equals $\mathbb{F}_q^n$.)

Generalizing a property of the generator matrix (2.2) noted above, encoding functions with the property that the symbols of the input word appear unchanged in some components of the codeword are known as *systematic encoders*. It is customary to call those components of the codewords the *information positions*. The remaining components of the codewords are called *parity checks*. Systematic encoders are sometimes desirable from a practical point of view because the information positions can be copied directly from the word to be encoded; only the parity checks need to be computed. There are corresponding savings in the decoding operation as well. If information is systematically encoded and no errors occur in transmission, the words in the message can be obtained directly from the received words by simply removing the parity checks. (It is perhaps worthwhile to mention again at this point that the goal of the encoding schemes we are considering here is *reliability* of information transmission, not secrecy!)

Exercise 6. Suppose that the generator matrix for a linear code C has the systematic form $G = (I_k \mid P)$, where I_k is a $k \times k$ identity matrix, and P is some $k \times (n - k)$ matrix. Show that

$$H = \begin{pmatrix} -P \\ I_{n-k} \end{pmatrix}$$

is a parity check matrix for C.

We will refer to a linear code with block length n, dimension k, and minimum distance d as an $[n, k, d]$ code. For instance, the Hamming code given by the generator matrix (2.2) is a [7,4,3] code.

Determining which triples of parameters $[n, k, d]$ can be realized by codes over a given finite field $\mathbb{F}_q$ and constructing such codes are two important problems in coding theory. These questions are directly motivated by the

decisions an engineer would need to make in selecting a code for a given application. Since an $[n, k, d]$ code has q^k distinct codewords, the choice of the parameter k will be determined by the size of the collection of words appearing in the messages to be transmitted. Based on the characteristics of the channel over which the transmission takes place (in particular the probability that an error occurs in transmission of a symbol), a value of d would be chosen to ensure that the probability of receiving a word that could not be correctly decoded was acceptably small. The remaining question would be how big to take n to ensure that a code with the desired parameters k and d actually exists. It is easy to see that, fixing k, we can construct codes with d as large as we like by taking n very large. (For instance, our codewords could consist of many concatenated copies of the corresponding words in $\mathbb{F}_q^k$.) However, the resulting codes would usually be too redundant to be practically useful. "Good" codes are ones for which the *information rate* $R = k/n$ is not too small, but for which d is relatively large. There is a famous result known as Shannon's Theorem (for the precise statement see, e.g., [vLi]) that ensures the existence of "good" codes in this sense, but the actual construction of "good" codes is one of the main problems in coding theory.

Exercise 7. In the following exercises, we explore some theoretical results giving various bounds on the parameters of codes. One way to try to produce good codes is to fix a block length n and a minimum distance d, then attempt to maximize k by choosing the codewords one by one so as to keep $d(x, y) \geq d$ for all distinct pairs $x \neq y$.

a. Show that $b = |B_{d-1}(c)|$ is given by $b = \sum_{i=0}^{d-1} \binom{n}{i}(q-1)^i$ for each $c \in \mathbb{F}_q^n$.

b. Let d be a given positive integer, and let C be a subset $C \subset \mathbb{F}_q^n$ (not necessarily a linear code) such that $d(x, y) \geq d$ for all pairs $x \neq y$ in C. Assume that for all $z \in \mathbb{F}_q^n \setminus C$, $d(z, c) \leq d - 1$ for some $c \in C$. Then show that $b \cdot |C| \geq q^n$ (b as in part a). This result gives one form of the *Gilbert-Varshamov bound*. Hint: An equivalent statement is that if $b \cdot |C| < q^n$, then there exists some z such that every pair of distinct elements in $C \cup \{z\}$ is still separated by at least d.

c. Show that if k satisfies $b < q^{n-k+1}$, then an $[n, k, d]$ linear code exists. Hint: By induction, we may assume that an $[n, k-1, d]$ linear code C exists. Using part b, consider the linear code C' spanned by C and z, where the distance from z to any word in C is $\geq d$. Show that C' still has minimum distance d.

d. On the other hand, show that for any linear code $d \leq n - k + 1$. This result is known as the *Singleton bound*. Hint: Consider what happens when a subset of $d-1$ components is deleted from each of the codewords.

Many other theoretical results, including both upper and lower bounds on the n, k, d parameters of codes, are also known. See the coding theory texts mentioned at the start of this section.

We now turn to the encoding and decoding operations. Our first observation is that encoding is much simpler to perform for linear codes than for arbitrary codes. For a completely arbitrary C of size q^k there would be little alternative to using some form of table look-up to compute the encoding function. On the other hand, for a linear code all the information about the code necessary for encoding is contained in the generator matrix (only k basis vectors for C rather than the whole set of q^k codewords), and all operations necessary for encoding may be performed using linear algebra.

Decoding a linear code is also correspondingly simpler. A general method, known as *syndrome decoding*, is based on the following observation. If $c = wG$ is a codeword, and some errors $e \in \mathbb{F}_q^n$ are introduced on transmission of c, the received word will be $x = c + e$. Then $cH = 0$ implies that $xH = (c + e)H = cH + eH = 0 + eH = eH$. Hence xH depends only on the error. The possible values for $eH \in \mathbb{F}_q^{n-k}$ are known as *syndromes*, and it is easy to see that the syndromes are in one-to-one correspondence with the cosets of C in $\mathbb{F}_q^n$ (or elements of the quotient space $\mathbb{F}_q^n/C \cong \mathbb{F}_q^{n-k}$), so there are exactly q^{n-k} of them. (See Exercise 12 below.)

Syndrome decoding works as follows. First, a preliminary calculation is performed, before any decoding. We construct a table, indexed by the possible values of the syndrome $s = xH$, of the element(s) in the corresponding coset with the smallest number of nonzero entries. These special elements of the cosets of C are called the *coset leaders*.

Exercise 8. Say $d = 2t + 1$, so we know that any error vector of weight t or less can be corrected. Show that if there are any elements of a coset of C which have t or fewer nonzero entries, then there is only one such element, and as a result the coset leader is unique.

If $x \in \mathbb{F}_q^n$ is received, we first compute the syndrome $s = xH$ and look up the coset leader(s) ℓ corresponding to s in our table. If there is a unique leader, we replace x by $x' = x - \ell$, which is in C (why?). (If $s = 0$, then $\ell = 0$, and $x' = x$ is itself a codeword.) Otherwise, we report an "error" value. By Exercise 8, if no more than t errors occurred in x, then we have found the unique codeword closest to the received word x and we return $E^{-1}(x')$. Note that by this method we have accomplished nearest neighbor decoding *without* computing $d(x, c)$ for all q^k codewords. However, a potentially large collection of information must be maintained to carry out this procedure—the table of coset leader(s) for each of the q^{n-k} cosets of C. In cases of practical interest, $n - k$ and q can be large, so q^{n-k} can be huge.

Exercise 9. Compute the table of coset leaders for the [7,4,3] Hamming code from (2.2). Use syndrome decoding to decode the received word $(1, 1, 0, 1, 1, 1, 0)$.

Here is another example of a linear code, this time over the field $\mathbb{F}_4 = \mathbb{F}_2[\alpha]/\langle \alpha^2 + \alpha + 1 \rangle$. Consider the code C with $n = 8$, $k = 3$ over $\mathbb{F}_4$ defined by the generator matrix:

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & \alpha & \alpha & \alpha^2 & \alpha^2 \\ 0 & 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha^2 \end{pmatrix}. \tag{2.4}$$

Note that G does not have the systematic form we saw above for the Hamming code's generator matrix. Though this is not an impediment to encoding, we can also obtain a systematic generator matrix for C by row-reduction (Gauss-Jordan elimination). This corresponds to changing basis in C; the image of the encoding map $E : \mathbb{F}_4^3 \to \mathbb{F}_4^8$ is not changed. It is a good exercise in finite field arithmetic to perform this computation by hand. It can also be done in Maple as follows. For simplicity, we will write a for α within Maple. To work in $\mathbb{F}_4$ we begin by defining `a` as a root of the polynomial $x^2 + x + 1$ as above.

```
alias(a=RootOf(x^2+x+1)):
```

The generator matrix G is entered as

```
m :=array(1..3, 1..8, [[1, 1, 1, 1, 1, 1, 1, 1],
    [0, 0, 1, 1, a, a, a^2, a^2], [0, 1, a, a^2, a, a^2, a, a^2]]) :
```

Then the command

```
mr := Gaussjord(m) mod 2;
```

will perform Gauss-Jordan elimination with coefficients treated as elements of $\mathbb{F}_4$. (Recall Maple's capitalization convention for unevaluated function calls, discussed in §1.) The result should be

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

Note that a^2 is replaced by its reduced form $a + 1$ everywhere here.

In the reduced matrix, the second row has five nonzero entries. Hence the minimum distance d for this code is ≤ 5. By computing all $4^3 - 1 = 63$ nonzero codewords, it can be seen that $d = 5$. It is often quite difficult to determine the exact minimum distance of a code (especially when the number of nonzero codewords, $q^k - 1$, is large).

To conclude this section, we will develop a relationship between the minimum distance of a linear code and the form of parity check matrices for the code.

(2.5) Proposition. *Let C be a linear code with parity check matrix H. If no collection of $\delta - 1$ distinct rows of H is a linearly dependent subset of $\mathbb{F}_q^{n-k}$, then the minimum distance d of C satisfies $d \geq \delta$.*

PROOF. We use the result of Exercise 4. Let $x \in C$ be a nonzero codeword. From the equation $xH = 0$ in $\mathbb{F}_q^{n-k}$, we see that the components of x are the coefficients in a linear combination of the rows of H summing to the zero vector. If no collection of $\delta - 1$ distinct rows is linearly dependent, then x must have at least δ nonzero entries. Hence $d \geq \delta$. □

Additional Exercises for §2

Exercise 10. Consider the formal inner product on $\mathbb{F}_q^n$ defined by

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$$

(a bilinear mapping from $\mathbb{F}_q^n \times \mathbb{F}_q^n$ to $\mathbb{F}_q$; there is no notion of positive-definiteness in this context). Given a linear code C, let

$$C^\perp = \{x \in \mathbb{F}_q^n : \langle x, y \rangle = 0 \text{ for all } y \in C\},$$

the subspace of $\mathbb{F}_q^n$ orthogonal to C. If C is k-dimensional, then $C^\perp$ is a linear code of block length n and dimension $n - k$ known as the *dual code* of C.

a. Let $G = (I_k \mid P)$ be a systematic generator matrix for C. Determine a generator matrix for $C^\perp$. How is this related to the parity check matrix for C? (Note on terminology: Many coding theory texts define a parity check matrix for a linear code to be the transpose of what we are calling a parity check matrix. This is done so that the rows of a parity check matrix will form a basis for the dual code.)
b. Find generator matrices and determine the parameters $[n, k, d]$ for the duals of the Hamming code from (2.2), and the code from (2.4).

Exercise 11. (The Hamming codes) Let q be a prime power, and let $m \geq 1$. We will call a set S of vectors in $\mathbb{F}_q^m$ a *maximal pairwise linearly independent subset* of $\mathbb{F}_q^m$ if S has the property that no two distinct elements of S are scalar multiples of each other, and if S is maximal with respect to inclusion. For each pair (q, m) we can construct linear codes C by taking a parity check matrix $H \in M_{n \times m}(\mathbb{F}_q)$ whose rows form a maximal pairwise linearly independent subset of $\mathbb{F}_q^m$, and letting $C \subset \mathbb{F}_q^n$ be the set of solutions of the system of linear equations $xH = 0$. For instance, with $q = 2$, we can take the rows of H to be all the nonzero vectors in $\mathbb{F}_2^k$ (in any order)—see (2.3) for the case $q = 2, k = 3$. The codes with these parity check matrices are called the Hamming codes.

a. Show that if S is a maximal pairwise linearly independent subset of $\mathbb{F}_q^m$, then S has exactly $(q^m - 1)/(q - 1)$ elements. (This is the same as the number of points of the projective space $\mathbb{P}^{m-1}$ over $\mathbb{F}_q$.)
b. What is the dimension k of a Hamming code defined by an $n \times m$ matrix H?

c. Write down a parity check matrix for a Hamming code with $q = 3$, $k = 2$.
d. Show that the minimum distance of a Hamming code is always 3, and discuss the error-detecting and error-correcting capabilities of these codes.
e. Show that all the Hamming codes are *perfect* codes (see Exercise 5 above).

Exercise 12. Let C be an $[n, k, d]$ linear code with parity check matrix H. Show that the possible values for $yH \in \mathbb{F}_q^{n-k}$ (the *syndromes*) are in one-to-one correspondence with the cosets of C in $\mathbb{F}_q^n$ (or elements of the quotient space $\mathbb{F}_q^n/C \cong \mathbb{F}_q^{n-k}$). Deduce that there are q^{n-k} different syndrome values.

§3 Cyclic Codes

In this section, we will consider several classes of linear codes with even more structure, and we will see how some of the algorithmic techniques in symbolic algebra we have developed can be applied to encode them. First we will consider the class of cyclic codes. Cyclic codes may be defined in several ways—the most elementary is certainly the following: A *cyclic code* is a linear code with the property that the set of codewords is closed under cyclic permutations of the components of vectors in $\mathbb{F}_q^n$. Here is a simple example.

In $\mathbb{F}_2^4$, consider the $[4, 2, 2]$ code C with generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \tag{3.1}$$

from Exercise 1 in §2. As we saw there, C contains 4 distinct codewords. The codewords $(0, 0, 0, 0)$ and $(1, 1, 1, 1)$ are themselves invariant under all cyclic permutations. The codeword $(1, 0, 1, 0)$ is not itself invariant: shifting one place to the left (or right) we obtain $(0, 1, 0, 1)$. But this *is* another codeword: $(0, 1, 0, 1) = (1, 1)G \in C$. Similarly, shifting $(0, 1, 0, 1)$ one place to the left or right, we obtain the codeword $(1, 0, 1, 0)$ again. It follows that the set C is closed under all cyclic shifts.

The property of invariance under cyclic permutations of the components has an interesting algebraic interpretation. Using the standard isomorphism between $\mathbb{F}_q^n$ and the vector space of polynomials of degree at most $n - 1$ with coefficients in $\mathbb{F}_q$:

$$(a_0, a_1, \ldots, a_{n-1}) \leftrightarrow a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$$

we may identify a cyclic code C with the corresponding collection of polynomials of degree $n-1$. The right cyclic shift which sends $(a_0, a_1, \ldots, a_{n-1})$ to $(a_{n-1}, a_0, a_1, \ldots, a_{n-2})$ is the same as the result of multiplying the poly-

nomial $a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ by x, then taking the remainder on division by $x^n - 1$.

Exercise 1. Show that multiplying the polynomial $p(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ by x, then taking the remainder on division by $x^n - 1$ yields a polynomial whose coefficients are the same as those of $p(x)$, but cyclically shifted one place to the right.

This suggests that when dealing with cyclic codes we should consider the polynomials of degree at most $n - 1$ as the elements of the quotient ring $R = \mathbb{F}_q[x]/\langle x^n - 1\rangle$. The reason is that multiplication of $f(x)$ by x followed by division gives the standard representative for the product $xf(x)$ in R. Hence, from now on we will consider cyclic codes as a vector subspaces of the ring R which are closed under multiplication by the coset of x in R. Now we make a key observation.

Exercise 2. Show that if a vector subspace $C \subset R$ is closed under multiplication by $[x]$, then it is closed under multiplication by *every coset* $[h(x)] \in R$.

Exercise 2 shows that cyclic codes have the defining property of ideals in a ring. We record this fact in the following proposition.

(3.2) Proposition. *Let $R = \mathbb{F}_q[x]/\langle x^n - 1\rangle$. A vector subspace $C \subset R$ is a cyclic code if and only if C is an ideal in the ring R.*

The ring R shares a nice property with its "parent" ring $\mathbb{F}_q[x]$.

(3.3) Proposition. *Each ideal $I \subset R$ is principal, generated by the coset of a single polynomial g of degree $n - 1$ or less. Moreover, g is a divisor of $x^n - 1$ in $\mathbb{F}_q[x]$.*

Proof. By the standard characterization of ideals in a quotient ring (see e.g. [CLO] Chapter 5, §2, Proposition 10), the ideals in R are in one-to-one correspondence with the ideals in $\mathbb{F}_q[x]$ containing $\langle x^n - 1\rangle$. Let J be the ideal corresponding to I. Since all ideals in $\mathbb{F}_q[x]$ are principal, J must be generated by some $g(x)$. Since $x^n - 1$ is in J, $g(x)$ is a divisor of $x^n - 1$ in $\mathbb{F}_q[x]$. The ideal $I = J/\langle x^n - 1\rangle$ is generated by the coset of $g(x)$ in R. □

Naturally enough, the polynomial g in Proposition (3.3) is called a *generator polynomial* for the cyclic code.

Exercise 3. Identifying the 4-tuple $(a, b, c, d) \in \mathbb{F}_2^4$ with $[a + bx + cx^2 + dx^3] \in R = \mathbb{F}_2[x]/\langle x^4 - 1\rangle$, show that the cyclic code in $\mathbb{F}_2^4$ with generator

matrix (3.1) can be viewed as the ideal generated by the coset of $g = 1+x^2$ in R. Find the codewords of the cyclic code with generator $1 + x$ in R.

The *Reed-Solomon codes* are one particularly interesting class of cyclic codes used extensively in applications. For example, a clever combination of two of these codes is used for error control in playback of sound recordings in the Compact Disc audio system developed by Philips in the early 1980's. They are attractive because they have good *burst error* correcting capabilities (see Exercise 15 below) and also because efficient decoding algorithms are available for them (see the next section). We will begin with a description of these codes via generator matrices, then show that they have the invariance property under cyclic shifts.

Choose a finite field $\mathbb{F}_q$ and consider codes of block length $n = q - 1$ constructed in the following way. Let α be a primitive element for $\mathbb{F}_q$ (see Theorem (1.9) of this chapter), fix $k < q$, and let $L_{k-1} = \{\sum_{i=0}^{k-1} a_i t^i : a_i \in \mathbb{F}_q\}$ be the vector space of polynomials of degree at most $k-1 < q-1$ in $\mathbb{F}_q[t]$. We make words in $\mathbb{F}_q^{q-1}$ by *evaluating* polynomials in L_{k-1} at the $q - 1$ nonzero elements of $\mathbb{F}_q$. By definition

$$C = \{(f(1), f(\alpha), \ldots, f(\alpha^{q-2})) \in \mathbb{F}_q^{q-1} : f \in L_{k-1}\} \tag{3.4}$$

is a Reed-Solomon code, sometimes denoted by $RS(k, q)$. C is a vector subspace of $\mathbb{F}_q^{q-1}$ since it is the image of the vector space L_{k-1} under the linear evaluation mapping

$$f \mapsto (f(1), f(\alpha), \ldots, f(\alpha^{q-2})).$$

Generator matrices for Reed-Solomon codes can be obtained by taking any basis of L_{k-1} and evaluating to form the corresponding codewords. The monomial basis $\{1, t, t^2, \ldots, t^{k-1}\}$ is the simplest. For example, consider the Reed-Solomon code over $\mathbb{F}_9$ with $k = 3$. Using the basis $\{1, t, t^2\}$ for L_3, we obtain the generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & 1 & \alpha^2 & \alpha^4 & \alpha^6 \end{pmatrix}, \tag{3.5}$$

where the first row gives the values of $f(t) = 1$, the second row gives the values of $f(t) = t$, and the third gives the values of $f(t) = t^2$ at the nonzero elements of $\mathbb{F}_9$ (recall, $\alpha^8 = 1$ in $\mathbb{F}_9$). For all $k < q$, the first k columns of the generator matrix corresponding to the monomial basis of L_{k-1} give a submatrix of Vandermonde form with nonzero determinant. It follows that the evaluation mapping is one-to-one, and the corresponding Reed-Solomon code is a linear code with block length $n = q - 1$, and dimension $k = \dim L_{k-1}$.

The generator matrix formed using the monomial basis of L_{k-1} also brings the cyclic nature of Reed-Solomon codes into sharp focus. Observe that each cyclic shift of a row of the matrix G in (3.5) yields a scalar

multiple of the same row. For example, cyclically shifting the third row one space to the right, we obtain

$$(\alpha^6, 1, \alpha^2, \alpha^4, \alpha^6, 1, \alpha^2, \alpha^4) = \alpha^6 \cdot (1, \alpha^2, \alpha^4, \alpha^6, 1, \alpha^2, \alpha^4, \alpha^6).$$

Exercise 4. Show that the other rows of (3.5) also have the property that a cyclic shift takes the row to a scalar multiple of the same row. Show that this observation implies this Reed-Solomon code is cyclic. Then generalize your arguments to *all* Reed-Solomon codes. Hint: Use the original definition of cyclic codes—closure under all cyclic shifts. You may wish to begin by showing that the cyclic shifts are linear mappings on $\mathbb{F}_q^n$.

We will give another proof that Reed-Solomon codes are cyclic below, and also indicate how to find the generator polynomial. However, we pause at this point to note one of the other interesting properties of Reed-Solomon codes. Since no polynomial in L_{k-1} can have more than $k-1$ zeroes in $\mathbb{F}_q$, every codeword in C has at least $(q-1)-(k-1) = q-k$ nonzero components (and some have exactly this many). By Exercise 4 of §2, the minimum distance for a Reed-Solomon code is $d = q-k = n-k+1$. Comparing this with the Singleton bound from part d of Exercise 7 from §2, we see that Reed-Solomon codes have the maximum possible d for the block length $q-1$ and dimension k. Codes with this property are called *MDS (maximum distance separable) codes* in the literature. So Reed-Solomon codes are good in this sense. However, their fixed, small block length relative to the size of the alphabet is sometimes a disadvantage. There is a larger class of cyclic codes known as *BCH codes* which contain the Reed-Solomon codes as a special case, but which do not have this limitation. Moreover, a reasonably simple lower bound on d is known for all BCH codes. See Exercise 13 below and [MS] or [vLi] for more on BCH codes.

Next, we will see another way to show that Reed-Solomon codes are cyclic that involves somewhat more machinery, but sheds additional light on the structure of cyclic codes of block length $q-1$ in general. Recall from Proposition (3.3) that the generator polynomial of a cyclic code of block length $q-1$ is a divisor of $x^{q-1}-1$. By Lagrange's Theorem, each of the $q-1$ nonzero elements of $\mathbb{F}_q$ is a root of $x^{q-1}-1=0$, hence

$$x^{q-1}-1 = \prod_{\beta \in \mathbb{F}_q^*} (x-\beta)$$

in $\mathbb{F}_q[x]$, where $\mathbb{F}_q^*$ is the set of nonzero elements of $\mathbb{F}_q$. Consequently, the divisors of $x^{q-1}-1$ are precisely the polynomials of the form $\prod_{\beta \in S}(x-\beta)$ for subsets $S \subset \mathbb{F}_q^*$. This is the basis for another characterization of cyclic codes.

Exercise 5. Show that a linear code of dimension k in $R = \mathbb{F}_q[x]/\langle x^{q-1}-1\rangle$ is cyclic if and only if the codewords, viewed as polynomials of degree at

most $q-2$, have some set S of $q-k-1$ common zeroes in $\mathbb{F}_q^*$. Hint: If the codewords have the elements in S as common zeroes, then each codeword is divisible by $g(x) = \prod_{\beta \in S}(x-\beta)$.

Using this exercise, we will now determine the generator polynomial of a Reed-Solomon code. Let $f(t) = \sum_{j=0}^{k-1} a_j t^j$ be an element of L_{k-1}. Consider the values $c_i = f(\alpha^i)$ for $i = 0, \ldots, q-2$. Using the c_i as the coefficients of a polynomial as in the discussion leading up to Proposition (3.2), write the corresponding codeword as $c(x) = \sum_{i=0}^{q-2} c_i x^i$. But then substituting for c_i and interchanging the order of summation, we obtain

$$\begin{aligned} c(\alpha^\ell) &= \sum_{i=0}^{q-2} c_i \alpha^{i\ell} \\ &= \sum_{j=0}^{k-1} a_j \left(\sum_{i=0}^{q-2} \alpha^{i(\ell+j)} \right). \end{aligned} \tag{3.6}$$

Assume that $1 \le \ell \le q-k-1$. Then for all $0 \le j \le k-1$, we have $1 \le \ell+j \le q-2$. By the result of Exercise 8 of §1, each of the inner sums on the right is zero so $c(\alpha^\ell) = 0$. Using Exercise 5, we have obtained another proof of the fact that Reed-Solomon codes are cyclic, since the codewords have the set of common zeroes $S = \{\alpha, \alpha^2, \ldots, \alpha^{q-k-1}\}$. Moreover, we have the following result.

(3.7) Proposition. *Let C be the Reed-Solomon code of dimension k and minimum distance $d = q-k$ over $\mathbb{F}_q$. Then the generator polynomial of C has the form*

$$g = (x-\alpha)\cdots(x-\alpha^{q-k-1}) = (x-\alpha)\cdots(x-\alpha^{d-1}).$$

For example, the Reed-Solomon codewords corresponding to the three rows of the matrix G in (3.5) above are $c_1 = 1 + x + x^2 + \cdots + x^7$, $c_2 = 1+\alpha x+\alpha^2 x^2+\cdots+\alpha^7 x^7$, and $c_3 = 1+\alpha^2 x+\alpha^4 x^2+\alpha^6 x^4+\cdots+\alpha^6 x^7$. Using Exercise 8 of §1, it is not difficult to see that the common roots of $c_1(x) = c_2(x) = c_3(x) = 0$ in $\mathbb{F}_9$ are $x = \alpha, \alpha^2, \ldots, \alpha^5$, so the generator polynomial for this code is

$$g = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5).$$

Also see Exercise 11 below for another point of view on Reed-Solomon and related codes.

From the result of Proposition (3.2), it is natural to consider the following generalization of the cyclic codes described above. Let R be a quotient ring of $\mathbb{F}_q[x_1, \ldots, x_m]$ of the form

$$R = \mathbb{F}_q[x_1, \ldots, x_m]/\langle x_1^{n_1} - 1, \ldots, x_m^{n_m} - 1 \rangle$$

for some $n_1, \ldots, n_m$. Any ideal I in R will be a linear code closed under products by arbitrary $h(x_1, \ldots, x_n)$ in R. We will call any code obtained in this way an *m-dimensional cyclic code*.

Note first that $\mathcal{H} = \{x_1^{n_1} - 1, \ldots, x_m^{n_m} - 1\}$ is a Gröbner basis for the ideal it generates, with respect to all monomial orders. (This follows for instance from Theorem 3 and Proposition 4 of Chapter 2, §9 of [CLO].) Hence standard representatives for elements of R can be computed by applying the division algorithm in $\mathbb{F}_q[x_1, \ldots, x_m]$ and computing remainders with respect to $\mathcal{H}$. We obtain in this way as representatives of elements of R all polynomials whose degree in x_i is $n_i - 1$ or less for each i.

Exercise 6. Show that as a vector space,

$$R = \mathbb{F}_q[x_1, \ldots, x_m]/\langle x_1^{n_1} - 1, \ldots, x_m^{n_m} - 1\rangle \cong \mathbb{F}_q^{n_1 \cdot n_2 \cdots n_m}.$$

Multiplication of an element of R by x_1, for example, can be viewed as a sort of cyclic shift in one of the variables. Namely, writing a codeword $c(x_1, \ldots, x_n) \in I$ as a polynomial in x_1, whose coefficients are polynomials in the other variables: $c = \sum_{j=0}^{n_1-1} c_j(x_2, \ldots, x_n)x_1^j$, multiplication by x_1, followed by division by $\mathcal{H}$ yields the standard representative $x_1 c = c_{n_1-1} + c_0 x_1 + c_1 x_1^2 + \cdots + c_{n_1-2} x_1^{n_1-1}$. Since $c \in I$ this shifted polynomial is also a codeword. The same is true for each of the other variables $x_2, \ldots, x_m$.

In the case $m = 2$, for instance, it is customary to think of the codewords of a 2-dimensional cyclic code either as polynomials in two variables, or as $n_1 \times n_2$ matrices of coefficients. In the matrix interpretation, multiplication by x_1 then corresponds to the right cyclic shift on each row, while multiplication by x_2 corresponds to a cyclic shift on each of the columns. Each of these operations leaves the set of codewords invariant.

Exercise 7. Writing $\mathbb{F}_4 = \mathbb{F}_2[\alpha]/\langle \alpha^2 + \alpha + 1\rangle$, the ideal $I \subset \mathbb{F}_4[x, y]/\langle x^3 - 1, y^3 - 1\rangle$ generated by $g_1(x, y) = x^2 + \alpha^2 xy + \alpha y$, $g_2(x, y) = y + 1$ gives an example of a 2-dimensional cyclic code with $n = 3^2 = 9$. As an exercise, determine k, the vector space dimension of this 2-dimensional cyclic code, by determining a vector space basis for I over $\mathbb{F}_4$. (Answer: $k = 7$. Also see the discussion following Theorem (3.9) below.) The minimum distance of this code is $d = 2$. Do you see why?

To define an m-dimensional cyclic code, it suffices to give a set of generators $\{[f_1], \ldots, [f_s]\} \subset R$ for the ideal $I \subset R$. The corresponding ideal J in $\mathbb{F}_q[x_1, \ldots, x_m]$ is

$$J = \langle f_1, \ldots, f_s\rangle + \langle x_1^{n_1-1} - 1, \ldots, x_m^{n_m-1} - 1\rangle.$$

Fix any monomial order on $\mathbb{F}_q[x_1, \ldots, x_m]$. With a Gröbner basis $G = \{g_1, \ldots, g_t\}$ for J with respect to this order we have everything necessary to

determine whether a given element of R is in I using the division algorithm in $\mathbb{F}_q[x_1, \ldots, x_m]$.

(3.8) Proposition. *Let R, I, J, G be as above. A polynomial $h(x_1, \ldots, x_n)$ represents an element of I in R if and only if its remainder on division by G is zero.*

PROOF. This follows because $I = J/\langle x_1^{n_1 - 1} - 1, \ldots, x_m^{n_m - 1} - 1\rangle$ and standard isomorphism theorems (see Theorem 2.6 of [Jac]) give a ring isomorphism

$$R/I \cong \mathbb{F}_q[x_1, \ldots, x_m]/J.$$

See Exercise 14 below for the details. □

An immediate consequence of Proposition (3.8) is the following systematic encoding algorithm for m-dimensional cyclic codes using division with respect to a Gröbner basis. One of the advantages of m-dimensional cyclic codes over linear codes in general is that their extra structure allows a very compact representation of the encoding function. We only need to know a reduced Gröbner basis for the ideal J corresponding to a cyclic code to perform systematic encoding. A Gröbner basis will generally have fewer elements than a vector space basis of I. This frequently means that much less information needs to be stored. In the following description of a systematic encoder, the *information positions* of a codeword will refer to the k positions in the codeword that duplicate the components of the element of $\mathbb{F}_q^k$ that is being encoded. These will correspond to a certain subset of the coefficients in a polynomial representative for an element of R. Similarly, the parity check positions are the complementary collection of coefficients.

(3.9) Theorem. *Let $I \subset R = \mathbb{F}_q[x_1, \ldots, x_m]/\langle x_1^{n_1} - 1, \ldots, x_m^{n_m} - 1\rangle$ be an m-dimensional cyclic code, and let G be a Gröbner basis for the corresponding ideal $J \subset \mathbb{F}_q[x_1, \ldots, x_m]$ with respect to some monomial order. Then there is a systematic encoding function for I constructed as follows.*

a. *The information positions are the coefficients of the nonstandard monomials for J in which each x_i appears to a power at most $n_i - 1$. (Non-standard monomials are monomials in $\langle \mathrm{LT}(J)\rangle$.)*
b. *The parity check positions are the coefficients of the standard monomials. (The standard monomials are those not contained in $\langle \mathrm{LT}(J)\rangle$.)*
c. *The following algorithm gives a systematic encoder E for I:*

Input: the Gröbner basis G for J,
w, a linear combination of nonstandard monomials
Output: $E(w) \in I$

Uses: Division algorithm with respect to given order

$\overline{w} := \overline{w}^G$ (the remainder on division)

$E(w) := w - \overline{w}$

PROOF. The dimension of R/I as a vector space over $\mathbb{F}_q$ is equal to the number of standard monomials for J since $R/I \cong \mathbb{F}_q[x_1, \ldots, x_m]/J$. (See for instance Proposition 4 from Chapter 5, §3 of [CLO].) The dimension of I as a vector space over $\mathbb{F}_q$ is equal to the difference $\dim R - \dim R/I$. But this is the same as the number of nonstandard monomials for J, in which each x_i appears to a power at most $n_i - 1$. Hence the span of those monomials is a subspace of R of the same dimension as I. Let w be a linear combination of only these nonstandard monomials. By the properties of the division algorithm, $\overline{w}$ is a linear combination of only standard monomials, so the symbols from w are not changed in the process of computing $E(w) = w - \overline{w}$. By Proposition (3.8), the difference $w - \overline{w}$ is an element of the ideal I, so it represents a codeword. As a result E is a systematic encoding function for I. □

In the case $m = 1$, the Gröbner basis for J is the generator polynomial g, and the remainder $\overline{w}$ is computed by ordinary 1-variable polynomial division. For example, let $\mathbb{F}_9 = \mathbb{F}_3[\alpha]/\langle \alpha^2 + \alpha + 2 \rangle$ (α is a primitive element by (1.1)) and consider the Reed-Solomon code over $\mathbb{F}_9$ with $n = 8$, $k = 5$. By Proposition (3.7), the generator polynomial for this code is $g = (x - \alpha)(x - \alpha^2)(x - \alpha^3)$, and $\{g\}$ is a Gröbner basis for the ideal J in $\mathbb{F}_9[x]$ corresponding to the Reed-Solomon code. By Theorem (3.9), as information positions for a systematic encoder we can take the coefficients of the nonstandard monomials $x^7, x^6, \ldots, x^3$ in an element of $\mathbb{F}_9[x]/\langle x^8 - 1 \rangle$. The parity check positions are the coefficients of the standard monomials $x^2, x, 1$. To encode a word $w(x) = x^7 + \alpha x^5 + (\alpha + 1)x^3$, for instance, we divide g into w, obtaining the remainder $\overline{w}$. Then $E(w) = w - \overline{w}$. Here is a Maple session performing this computation. We use the method discussed in §§1,2 for dealing with polynomials with coefficients in a finite field. First we find the generator polynomial for the Reed-Solomon code as above, using:

```
alias(alpha = RootOf(t^2 + t + 2));
g := collect(Expand((x-alpha)*(x-alpha^2)*
   (x-alpha^3) mod 3,x);
```

This produces output

$$g := x^3 + alpha\ x^2 + (1 + alpha)x + 2\ alpha + 1.$$

Then

```
w := x^7 + alpha*x^5 + (alpha + 1)*x^3:
(w - Rem(w,g,x)) mod 3;
```

yields output as follows

$$x^7 + alpha\ x^5 + (1 + alpha)x^3 + 2(2 + 2\ alpha)x^2 + x + 2.$$

After simplifying the coefficient of x^2 to $\alpha + 1$, this is the Reed-Solomon codeword.

Next, we consider the 2-dimensional cyclic code in Exercise 7. Recall $I \subset R = \mathbb{F}_4[x, y]/\langle x^3 - 1, y^3 - 1\rangle$ generated by $g_1(x, y) = x^2 + \alpha^2 xy + \alpha y$, $g_2(x, y) = y + 1$. Take $\mathbb{F}_4 = \mathbb{F}_2[\alpha]/\langle \alpha^2 + \alpha + 1\rangle$ and note that $-1 = +1$ in this field. Hence $x^3 - 1$ is the same as $x^3 + 1$, and so forth. As above, we must consider the corresponding ideal

$$J = \langle x^2 + \alpha^2 xy + \alpha y, y + 1, x^3 + 1, y^3 + 1\rangle$$

in $\mathbb{F}_4[x, y]$. Applying Buchberger's algorithm to compute a reduced *lex* Gröbner basis ($x > y$) for this ideal, we find

$$G = \{x^2 + \alpha^2 x + \alpha, y + 1\}.$$

As an immediate result, the quotient ring $\mathbb{F}_4[x, y]/J \cong R/I$ is 2-dimensional, while R is 9-dimensional over $\mathbb{F}_4$. Hence I has dimension $9 - 2 = 7$. There are also exactly two points in $\mathbf{V}(J)$. According to Theorem (3.9), the information positions for this code are the coefficients of $x^2, y, xy, x^2y, y^2, xy^2, x^2y^2$, and the parity checks are the coefficients of $1, x$. To encode $w = x^2y^2$ for example, we would compute the remainder on division by G, which is $\overline{x^2y^2}^G = \alpha^2 x + \alpha$ then subtract to obtain $E(w) = x^2y^2 + \alpha^2 x + \alpha$.

Gröbner basis computations in polynomial rings over finite fields may be done with Maple's `Groebner` and `Ore_algebra` packages as follows. For example, to compute the example above, we would first load the `Ore_algebra` and `Groebner` packages, then define the polynomial ring using

```
A:= poly_algebra(x,y,a,characteristic=2, alg_relations={a^2+a+1});
```

(This defines a ring A which is isomorphic to $\mathbb{F}_4[x, y]$. Here a is the primitive element for $\mathbb{F}_4$ and the idea is the same as in our earlier computations with a variable aliased as a root of a given irreducible polynomial. However, that method and the `mod` environment are not compatible with the Gröbner basis routines.) Then define the *lex* order as follows.

```
TL:=termorder(A,plex(x,y,a));
```

(Note that a is included.) Finally, if we declare

```
J:=[x^2+a^2*x*y+a*y,y+1,x^3+1,y^3+1];
```

then the command

```
gbasis(J,TL);
```

will do the Gröbner basis computation in the ring A. Other computer algebra systems such as `Singular` and *Macaulay 2* can handle these computations.

Additional Exercises for §3

Exercise 8. Let C be a cyclic code in $R = \mathbb{F}_q[x]/\langle x^n - 1\rangle$, with monic generator polynomial $g(x)$ of degree $n - k$, so that the dimension of C is k. Write out a generator matrix for C as a linear code, viewing the encoding procedure of Theorem (3.9) as a linear map from the span of $\{x^{n-k}, x^{n-k+1}, \dots, x^{n-1}\}$ to R. In particular show that every row of the matrix is determined by the first row, i.e. the image $E(x^{n-k})$. This gives another way to understand how the cyclic property reduces the amount of information necessary to describe a code.

Exercise 9. This exercise will study the dual of a cyclic code of block length $q - 1$ or $(q - 1)^m$ more generally. See Exercise 10 from §2 for the definition of the dual of a linear code. Let $R = \mathbb{F}_q[x]/\langle x^{q-1} - 1\rangle$ as in the discussion of Reed-Solomon codes.

a. Show that if $f(x) = \sum_{i=0}^{q-2} a_i x^i$ and $h(x) = \sum_{i=0}^{q-2} b_i x^i$ represent any two elements of R, then the inner product $\langle a, b\rangle$ of their vectors of coefficients is the same as the constant term in the product $f(x)h(x^{-1}) = f(x)h(x^{q-2})$ in R.
b. Let C be a cyclic code in R. Show that the dual code $C^\perp$ is equal to the collection of polynomials $h(x)$ such that $f(x)h(x^{-1}) = 0$ (product in R) for all $f(x) \in C$.
c. Use part b to describe the generator polynomial for $C^\perp$ in terms of the generator $g(x)$ for C. Hint: recall from the proof of Proposition (3.3) that $g(x)$ is a divisor of $x^{q-1} - 1 = \prod_{\beta \in \mathbb{F}_q^*}(x - \beta)$. The generator polynomial for $C^\perp$ will have the same property.
d. Extend these results to m-dimensional cyclic codes in

$$\mathbb{F}_q[x_1, \dots, x_m]/\langle x_i^{q-1} - 1 : i = 1, \dots, m\rangle.$$

Exercise 10. This exercise discusses another approach to the study of cyclic codes of block-length $q - 1$, which recovers the result of Exercise 5 in a different way. Namely, consider the ring $R = \mathbb{F}_q[x]/\langle x^{q-1} - 1\rangle$. The structure of the ring R and its ideals may be studied as follows.

a. Show that

$$\begin{aligned} \varphi : R &\to \mathbb{F}_q^{q-1} \\ c(x) &\mapsto (c(1), c(\alpha), \dots, c(\alpha^{q-2})) \end{aligned} \tag{3.10}$$

defines a bijective mapping, which becomes an *isomorphism of rings* if we introduce the component-wise product

$$(c_0, \dots, c_{q-2}) \cdot (d_0, \dots, d_{q-2}) = (c_0 d_0, \dots, c_{q-2} d_{q-2})$$

as multiplication operation in $\mathbb{F}_q^{q-1}$. (The mapping φ is a discrete analogue of the *Fourier transform* since it takes polynomial products in

R—convolution on the coefficients—to the component-wise products in $\mathbb{F}_q^{q-1}$.)

b. Show that the ideals in the ring $\mathbb{F}_q^{q-1}$ (with the component-wise product) are precisely the subsets of the following form. For each collection of subscripts $S \subset \{0, 1, \ldots, q-2\}$, let

$$I_S = \{(c_0, \ldots, c_{q-2}) : c_i = 0 \text{ for all } i \in S\}.$$

Then each ideal is equal to I_S for some S.

c. Using the mapping φ, deduce from part b and Proposition (3.2) that cyclic codes in R are in one-to-one correspondence with subsets $S \subset \{0, 1, \ldots, q-2\}$, or equivalently subsets of the nonzero elements of the field, $\mathbb{F}_q^*$. Given a cyclic code $C \subset R$, the corresponding subset of $\mathbb{F}_q^*$ is called the set of *zeroes* of C. For Reed-Solomon codes the set of zeroes has the form $\{\alpha, \ldots, \alpha^{q-k-1}\}$ (a "consecutive string" of zeroes starting from α).

Exercise 11.

a. By constructing an appropriate *transform* φ analogous to the map in (3.10), or otherwise, show that the results of Exercise 10 may be modified suitably to cover the case of m-dimensional cyclic codes of block length $n = (q-1)^m$. In particular, an m-dimensional cyclic code I in $\mathbb{F}_q[x_1, \ldots, x_m]/\langle x_1^{q-1} - 1, \ldots, x_m^{q-1} - 1\rangle$ is uniquely specified by giving a set of zeroes—the points of $\mathbf{V}(J)$—in $(\mathbb{F}_q^*)^m = \mathbf{V}(x_1^{q-1} - 1, \ldots, x_m^{q-1} - 1)$. (Readers of Chapter 2 should compare with the discussion of finite-dimensional algebras in §2 of that chapter.)

b. Consider the 2-dimensional cyclic code I in $\mathbb{F}_9[x, y]/\langle x^8 - 1, y^8 - 1\rangle$ generated by $g(x, y) = x^7y^7 + 1$. What is the dimension of I (i.e., the parameter k)? What is the corresponding set of zeroes in $(\mathbb{F}_9^*)^2$?

Exercise 12. In this exercise, we will explore the relation between the zeroes of a cyclic code and its minimum distance. Let α be a primitive element of $\mathbb{F}_q$. Consider a cyclic code C of length $q-1$ over $\mathbb{F}_q$ and suppose that there exist ℓ and $\delta \geq 2$ such that the $\delta - 1$ consecutive powers of α:

$$\alpha^\ell, \alpha^{\ell+1}, \ldots, \alpha^{\ell+\delta-2}$$

are distinct roots of the generator polynomial of C.

a. By considering the equations $c(\alpha^{\ell+j}) = 0$, $j = 0, \ldots, \delta - 2$, satisfied by the codewords (written as polynomials), show that the vectors

$$(1, \alpha^{\ell+j}, \alpha^{2(\ell+j)}, \ldots, \alpha^{(q-2)(\ell+j)}),$$

can be taken as columns of a parity check matrix H for C.

b. Show that, possibly after removing common factors from the rows, all the determinants of the $(\delta-1) \times (\delta-1)$ submatrices of H formed using entries in these columns are Vandermonde determinants.

c. Using Proposition (2.5), show that the minimum distance d of C satisfies $d \geq \delta$.
d. Use the result of part c to rederive the minimum distance of a Reed-Solomon code.

Exercise 13. (The BCH codes) Now consider cyclic codes C of length $q^m - 1$ over $\mathbb{F}_q$ for some $m \geq 1$.
a. Show that the result of Exercise 12 extends in the following way. Let α be a primitive element of $\mathbb{F}_{q^m}$, and suppose that there exist ℓ and $\delta \geq 2$ such that the $\delta - 1$ consecutive powers of α:

$$\alpha^\ell, \alpha^{\ell+1}, \ldots, \alpha^{\ell+\delta-2}$$

are distinct roots of the generator polynomial $g(x) \in \mathbb{F}_q[x]$ of C. Show that C has minimum distance $d \geq \delta$.
b. The "narrow-sense" q-ary BCH code $BCH_q(m,t)$ is the cyclic code over $\mathbb{F}_q$ whose generator polynomial is the *least common multiple* of the minimal polynomials of $\alpha, \alpha^2, \ldots, \alpha^{2t} \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$. (The minimal polynomial of $\beta \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is the nonzero polynomial of minimal degree in $\mathbb{F}_q[u]$ with β as a root.) Show the the minimum distance of $BCH_q(m,t)$ is at least $2t+1$. (The integer $2t+1$ is called the *designed distance* of the BCH code.)
c. Construct the generator polynomial for $BCH_3(2,2)$ (a code over $\mathbb{F}_3$). What is the dimension of this code?
d. Is it possible for the actual minimum distance of a BCH code to be strictly larger than its designed distance? For example, show using Proposition (2.5) that the actual minimum distance of the binary BCH code $BCH_2(5,4)$ satisfies $d \geq 11$ even though the designed distance is only 9. Hint: Start by showing that if $\beta \in \mathbb{F}_{2^m}$ is a root of a polynomial $p(u) \in \mathbb{F}_2[u]$, then so are $\beta^2, \beta^4, \ldots, \beta^{2^{m-1}}$. Readers familiar with Galois theory for finite fields will recognize that we are applying the *Frobenius automorphism* of $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$ from Exercise 14 of §1 repeatedly here.

Exercise 14. Prove Proposition (3.8).

Exercise 15. Reed-Solomon codes are now commonly used in situations such as communication to and from deep-space exploration craft, the CD digital audio system, and many others where errors tend to occur in "bursts" rather than randomly. One reason is that Reed-Solomon codes over an alphabet $\mathbb{F}_{2^r}$ with $r > 1$ can correct relatively long bursts of errors on the bit level, even if the minimum distance d is relatively small. Each Reed-Solomon codeword may be represented as a string of $(2^r - 1)r$ bits, since each symbol from $\mathbb{F}_{2^r}$ is represented by r bits. Show that a burst of $r\ell$ consecutive bit errors will change at most $\ell + 1$ of the entries of the

codeword, viewed as elements of $\mathbb{F}_{2^r}$. So if $\ell + 1 \leq \lfloor (d-1)/2 \rfloor$, a burst error of length $r\ell$ can be corrected. Compare with Proposition (2.1).

§4 Reed-Solomon Decoding Algorithms

The *syndrome decoding* method that we described in §2 can be applied to decode any linear code. However, as noted there, for codes with large codimension $n - k$, a very large amount of information must be stored to carry it out. In this section, we will see that there are much better methods available for the Reed-Solomon codes introduced in §3—methods which exploit their extra algebraic structure. Several different but related decoding algorithms for these codes have been considered. One well-known method is due to Berlekamp and Massey (see [Bla]). With suitable modifications, it also applies to the larger class of BCH codes mentioned in §3, and it is commonly used in practice. Other algorithms paralleling the Euclidean algorithm for the GCD of two polynomials have also been considered. Our presentation will follow two papers of Fitzpatrick ([Fit1], [Fit2]) which show how Gröbner bases for modules over polynomial rings (see Chapter 5) can be used to give a framework for the computations involved. Decoding algorithms for m-dimensional cyclic codes using similar ideas have been considered by Sakata ([Sak]), Heegard-Saints ([HeS]) and others.

To begin, we introduce some notation. We fix a field $\mathbb{F}_q$ and a primitive element α, and consider the Reed-Solomon code $C \subset \mathbb{F}_q/\langle x^{q-1} - 1 \rangle$ given by a generator polynomial

$$g = (x - \alpha) \cdots (x - \alpha^{d-1})$$

of degree $d - 1$. By Proposition (3.7), we know that the dimension of C is $k = q - d$, and the minimum distance of C is d. For simplicity we will assume that d is odd: $d = 2t + 1$. Then by Proposition (2.1), any error vector of weight t or less should be correctable.

Let $c = \sum_{j=0}^{q-2} c_j x^j$ be a codeword of C. Since C has generator polynomial $g(x)$, this means that in $\mathbb{F}_q[x]$, c is divisible by g. Suppose that c is transmitted, but some errors are introduced, so that the received word is $y = c + e$ for some $e = \sum_{i \in I} e_i x^i$. I is called the set of *error locations* and the coefficients e_i are known as the *error values*. To decode, we must solve the following problem.

(4.1) Problem. *Given a received word y, determine the set of error locations I and the error values e_i. Then the decoding function will return $E^{-1}(y - e)$.*

The set of values $E_j = y(\alpha^j)$, $j = 1, \ldots, d-1$, serves the same purpose as the syndrome of the received word for a general linear code. (It is not the same thing though—the direct analog of the syndrome would be the

remainder on division by the generator. See Exercise 7 below.) First, we can determine whether errors have occurred by computing the values E_j. If $E_j = y(\alpha^j) = 0$ for all $j = 1, \ldots, d-1$, then y is divisible by g. Assuming the error vector has a weight at most t, y must be the codeword we intended to send. If some $E_j \neq 0$, then there are errors and we can try to use the information included in the E_j to solve Problem (4.1). Note that the E_j are the values of the error polynomial for $j = 1, \ldots, d-1$:

$$E_j = y(\alpha^j) = c(\alpha^j) + e(\alpha^j) = e(\alpha^j),$$

since c is a multiple of g. (As in Exercise 10 from §3, we could also think of the E_j as a portion of the *transform* of the error polynomial.) The polynomial

$$S(x) = \sum_{j=1}^{d-1} E_j x^{j-1}$$

is called the *syndrome polynomial* for y. Its degree is $d-2$ or less. By extending the definition of $E_j = e(\alpha^j)$ to *all* exponents j we can also consider the formal power series

$$E(x) = \sum_{j=1}^{\infty} E_j x^{j-1}. \tag{4.2}$$

(Since $\alpha^q = \alpha$, the coefficients in E are periodic, with period at most q, and consequently E is actually the series expansion of a rational function of x; see (4.3) below. One can also solve the decoding problem by finding the recurrence relation of minimal order on the coefficients in E. For the basics of this approach see Exercise 6 below.)

Suppose we knew the error polynomial e. Then

$$E_j = \sum_{i \in I} e_i (\alpha^j)^i = \sum_{i \in I} e_i (\alpha^i)^j.$$

By expanding in formal geometric series, $E(x)$ from (4.2) can be written as

$$\begin{aligned} E(x) &= \sum_{i \in I} \frac{e_i \alpha^i}{(1 - \alpha^i x)} \\ &= \frac{\Omega(x)}{\Lambda(x)}, \end{aligned} \tag{4.3}$$

where

$$\Lambda = \prod_{i \in I} (1 - \alpha^i x)$$

and

$$\Omega = \sum_{i \in I} e_i \alpha^i \prod_{\substack{j \neq i \\ j \in I}} (1 - \alpha^j x).$$

The roots of Λ are precisely the α^{-i} for $i \in I$. Since the error locations can be determined easily from these roots, we call Λ the *error locator polynomial*. Turning to the numerator Ω, we see that

$$\deg(\Omega) \leq \deg(\Lambda) - 1.$$

In addition,

$$\Omega(\alpha^{-i}) = e_i \alpha^i \prod_{j \neq i, j \in I} (1 - \alpha^j \alpha^{-i}) \neq 0.$$

Hence Ω has no roots in common with Λ. From this we deduce the important observation that the polynomials Ω and Λ must be *relatively prime*.

Similarly, if we consider the "tail" of the series E,

$$\begin{aligned} E(x) - S(x) &= \sum_{j=d}^{\infty} \left(\sum_{i \in I} e_i (\alpha^i)^j \right) x^{j-1} \\ &= x^{d-1} \cdot \frac{\Gamma(x)}{\Lambda(x)}, \end{aligned} \tag{4.4}$$

where

$$\Gamma = \sum_{i \in I} e_i \alpha^{id} \prod_{\substack{j \neq i \\ j \in I}} (1 - \alpha^j x).$$

The degree of Γ is also at most $\deg(\Lambda) - 1$.

Combining (4.3) and (4.4), and writing $d - 1 = 2t$ we obtain the relation

$$\Omega = \Lambda S + x^{2t} \Gamma. \tag{4.5}$$

For some purposes, it will be more convenient to regard (4.5) as a *congruence*. The equation (4.5) implies that

$$\Omega \equiv \Lambda S \bmod x^{2t}. \tag{4.6}$$

Conversely, if (4.6) holds, there is some polynomial Γ such that (4.5) holds. The congruence (4.6), or sometimes its explicit form (4.5), is called the *key equation* for decoding.

The derivation of the key equation (4.6) assumed e was known. But now consider the situation in an actual decoding problem, assuming an error vector of weight at most t. Given the received word y, S is computed. The key equation (4.6) is now viewed as a relation between the known

polynomials S, x^{2t}, and the *unknowns* Ω, Λ. Suppose a solution $(\overline{\Omega}, \overline{\Lambda})$ of the key equation is found, which satisfies the following *degree conditions*:

$$\begin{cases} \deg(\overline{\Lambda}) \leq t \\ \deg(\overline{\Omega}) < \deg(\overline{\Lambda}) \end{cases} \tag{4.7}$$

and in which $\overline{\Omega}, \overline{\Lambda}$ are relatively prime. We claim that in such a solution $\overline{\Lambda}$ must be a factor of $x^{q-1} - 1$, and its roots give the inverses of the error locations. This is a consequence of the following uniqueness statement.

(4.8) Theorem. *Let S be the syndrome polynomial corresponding to a received word y with an error of weight at most t. Up to a constant multiple, there exists a unique solution (Ω, Λ) of (4.6) that satisfies the degree conditions (4.7), and in which Ω and Λ are relatively prime.*

Proof. As above, the actual error locator Λ and the corresponding Ω give one such solution. Let $(\overline{\Omega}, \overline{\Lambda})$ be any other. From the congruences

$$\overline{\Omega} \equiv \overline{\Lambda} S \bmod x^{2t}$$
$$\Omega \equiv \Lambda S \bmod x^{2t},$$

multiplying the second by $\overline{\Lambda}$, the first by Λ and subtracting, we obtain

$$\overline{\Omega}\Lambda \equiv \Omega\overline{\Lambda} \bmod x^{2t}.$$

Since the degree conditions (4.7) are satisfied for both solutions, both sides of this congruence are actually polynomials of degree at most $2t - 1$, so it follows that

$$\overline{\Omega}\Lambda = \Omega\overline{\Lambda}.$$

Since Λ and Ω are relatively prime, and similarly for $\overline{\Lambda}$ and $\overline{\Omega}$, Λ must divide $\overline{\Lambda}$ and vice versa. Similarly for Ω and $\overline{\Omega}$. As a result, Λ and $\overline{\Lambda}$ differ at most by a constant multiple. Similarly for Ω and $\overline{\Omega}$, and the constants must agree. □

Given a solution of (4.6) for which the conditions of Theorem (4.8) are satisfied, working backwards, we can determine the roots of $\overline{\Lambda} = 0$ in $\mathbb{F}_q^*$, and hence the error locations—if α^{-i} appears as a root, then $i \in I$ is an error location. Finally, the error values can be determined by the following observation.

Exercise 1. Let (Ω, Λ) be the solution of (4.6) in which the actual error locator polynomial Λ (with constant term 1) appears. If $i \in I$, show that

$$\Omega(\alpha^{-i}) = \alpha^i e_i \chi_i(\alpha^{-i}),$$

where $\chi_i = \prod_{j \neq i}(1 - \alpha^j x)$. Hence we can solve for e_i, knowing the error locations. The resulting expression is called the *Forney formula* for the error value.

Theorem (4.8) and the preceding discussion show that solving the decoding problem (4.1) can be accomplished by solving the key equation (4.6). It is here that the theory of module Gröbner bases can be applied to good effect. Namely, given the integer t and $S \in \mathbb{F}_q[x]$, consider the set of *all pairs* $(\Omega, \Lambda) \in \mathbb{F}_q[x]^2$ satisfying (4.6):

$$K = \{(\Omega, \Lambda) : \Omega \equiv \Lambda S \bmod x^{2t}\}.$$

Exercise 2. Show that K is a $\mathbb{F}_q[x]$-submodule of $\mathbb{F}_q[x]^2$. Also show that every element of K can be written as a combination (with polynomial coefficients) of the two generators

$$g_1 = (x^{2t}, 0) \text{ and } g_2 = (S, 1). \tag{4.9}$$

Hint: For the last part it may help to consider the related module

$$\overline{K} = \{(\Omega, \Lambda, \Gamma) : \Omega = \Lambda S + x^{2t}\Gamma\}$$

and the elements $(\Omega, \Lambda, \Gamma) = (x^{2t}, 0, 1), (S, 1, 0)$ in $\overline{K}$.

The generators for K given in (4.9) involve only the *known polynomials* for the decoding problem with syndrome S. Following Fitzpatrick, we will now show that (4.9) is a *Gröbner basis* for K with respect to one monomial order on $\mathbb{F}_q[x]^2$. Moreover, one of the special solutions $(\Lambda, \Omega) \in K$ given by Theorem (4.8) is guaranteed to occur in a Gröbner basis for K with respect to a second monomial order on $\mathbb{F}_q[x]^2$. These results form the basis for *two different* decoding methods that we will indicate.

To prepare for this, we need to begin by developing some preliminary facts about submodules of $\mathbb{F}_q[x]^2$ and monomial orders. The situation here is very simple compared to the general situation studied in Chapter 5. We will restrict our attention to submodules $M \subset \mathbb{F}_q[x]^2$ such that the quotient $\mathbb{F}_q[x]^2/M$ is *finite-dimensional* as a vector space over $\mathbb{F}_q$. We will see below that this is always the case for the module K with generators as in (4.9). There is a characterization of these submodules that is very similar to the Finiteness Theorem for quotients $k[x_1, \ldots, x_n]/I$ from Chapter 2, §2.

(4.10) Proposition. *Let k be any field, and let M be a submodule of $k[x]^2$. Let $>$ be any monomial order on $k[x]^2$. Then the following conditions are equivalent:*

a. *The k-vector space $k[x]^2/M$ is finite-dimensional.*
b. *$\langle \text{LT}_>(M) \rangle$ contains elements of the form $x^u\mathbf{e}_1 = (x^u, 0)$ and $x^v\mathbf{e}_2 = (0, x^v)$ for some $u, v \geq 0$.*

PROOF. Let $\mathcal{G}$ be a Gröbner basis for M with respect to the monomial order $>$. As in the ideal case, the elements of $k[x]^2/M$ are linear combinations of monomials in the complement of $\langle \text{LT}_>(M) \rangle$. There is a finite number

of such monomials if and only if $\langle \text{LT}_>(M)\rangle$ contains multiples of both $\mathbf{e}_1$ and $\mathbf{e}_2$. □

Every submodule we consider from now on in this section will satisfy the equivalent conditions in (4.10), even if no explicit mention is made of that fact.

The monomial orders that come into play in decoding are special cases of *weight* orders on $\mathbb{F}_q[x]^2$. They can also be described very simply "from scratch" as follows.

(4.11) Definition. Let $r \in \mathbb{Z}$, and define an order $>_r$ by the following rules. First, $x^m\mathbf{e}_i >_r x^n\mathbf{e}_i$ if $m > n$ and $i = 1$ or 2. Second, $x^m\mathbf{e}_2 >_r x^n\mathbf{e}_1$ if and only if $m + r \geq n$.

For example, with $r = 2$, the monomials in $k[x]^2$ are ordered by $>_2$ as follows:

$$\mathbf{e}_1 <_2 x\mathbf{e}_1 <_2 x^2\mathbf{e}_1 <_2 \mathbf{e}_2 <_2 x^3\mathbf{e}_1 <_2 x\mathbf{e}_2 <_2 x^4\mathbf{e}_1 <_2 \cdots .$$

Exercise 3.
a. Show that $>_r$ defines a monomial order on $k[x]^2$ for each $r \in \mathbb{Z}$.
b. How are the monomials in $k[x]^2$ ordered under $>_{-2}$?
c. Show that the $>_0$ and $>_{-1}$ orders coincide with TOP (*term over position*) orders as introduced in Chapter 5 (for different orderings of the standard basis).
d. Are the POT (*position over term*) orders special cases of the $>_r$ orders? Why or why not?

Gröbner bases for submodules with respect to the $>_r$ orders have very special forms.

(4.12) Proposition. *Let M be a submodule of $k[x]^2$, and fix $r \in \mathbb{Z}$. Assume $\langle \text{LT}_{>_r}(M)\rangle$ is generated by $x^u\mathbf{e}_1 = (x^u, 0)$ and $x^v\mathbf{e}_2 = (0, x^v)$ for some $u, v \geq 0$. Then a subset $\mathcal{G} \subset M$ is a reduced Gröbner basis of M with respect to $>_r$ if and only if $\mathcal{G} = \{g_1 = (g_{11}, g_{12}), g_2 = (g_{21}, g_{22})\}$, where the g_i satisfy the following two properties:*
a. $\text{LT}(g_1) = x^u\mathbf{e}_1$ *(in g_{11}), and* $\text{LT}(g_2) = x^v\mathbf{e}_2$ *(in g_{22}) for u, v as above.*
b. $\deg(g_{21}) < u$ *and* $\deg(g_{12}) < v$.

PROOF. Suppose $\mathcal{G}$ is a subset of M satisfying conditions a,b. By a, the leading terms of the elements of $\mathcal{G}$ generate $\langle \text{LT}(M)\rangle$, so by definition $\mathcal{G}$ is a Gröbner basis for M. Condition b implies that no terms in g_1 can be removed by division with respect to g_2 and vice versa, so $\mathcal{G}$ is reduced. Conversely, if $\mathcal{G}$ is a reduced Gröbner basis for M with respect to $>_r$ it must contain exactly two elements. Numbering the generators g_1 and g_2 as above condition a must hold. Finally b must hold if $\mathcal{G}$ is reduced. (Note,

fixing the leading terms in g_1 and g_2 implies that the other components satisfy $\deg(g_{12}) + r < u$ and $\deg(g_{21}) \leq v + r$.) □

An immediate, but important, consequence of Proposition (4.12) is the following observation.

(4.13) Corollary. *Let $\mathcal{G} = \{(S, 1), (x^{2t}, 0)\}$ be the generators for the module K of solutions of the key equation in the decoding problem with syndrome S. Then $\mathcal{G}$ is a Gröbner basis for K with respect to the order $>_{\deg(S)}$.*

Note $\text{LT}_{>_{\deg(S)}}((S, 1)) = (0, 1) = \mathbf{e}_2$, so the module of solutions of the key equation always satisfies the finiteness condition from Proposition (4.10). We leave the proof of Corollary (4.13) as an exercise for the reader.

The final general fact we will need to know is another consequence of the definition of a Gröbner basis. First we introduce some terminology.

(4.14) Definition. Let M be a nonzero submodule of $k[x]^2$. A *minimal element* of M with respect to a monomial order $>$ is a $g \in M \setminus \{0\}$ such that $\text{LT}(g)$ is minimal with respect to $>$.

For instance, from (4.13), $(S, 1)$ is minimal with respect to the order $>_{\deg(S)}$ in $\langle (S, 1), (x^{2t}, 0)\rangle$ since

$$\mathbf{e}_2 = \text{LT}((S, 1)) <_{\deg(S)} \text{LT}((x^{2t}, 0)) = x^{2t}\mathbf{e}_1,$$

and these leading terms generate $\langle \text{LT}(K)\rangle$ for the $>_{\deg(S)}$ order.

Exercise 4. Show that minimal elements of $M \subset k[x]^2$ are *unique*, up to a nonzero constant multiple.

As in the example above, once we fix an order $>_r$, a minimal element for M with respect to that order is *guaranteed* to appear in a Gröbner basis for M with respect to $>_r$.

(4.15) Proposition. *Fix any $>_r$ order on $k[x]^2$, and let M be a submodule. Every Gröbner basis for M with respect to $>_r$ contains a minimal element of M with respect to $>_r$.*

We leave the easy proof to the reader. Now we come to the main point. The special solution of the key equation (4.6) guaranteed by Theorem (4.8) can be characterized as the minimal element of the module K with respect to a suitable order.

(4.16) Proposition. *Let $g = (\overline{\Omega}, \overline{\Lambda})$ be a solution of the key equation satisfying the degree conditions (4.7) and with components relatively prime*

(which is unique up to constant multiple by Theorem (4.8)). Then g is a minimal element of K under the $>_{-1}$ order.

PROOF. An element $\overline{g} = (\overline{\Omega}, \overline{\Lambda}) \in K$ satisfies $\deg(\overline{\Lambda}) > \deg(\overline{\Omega})$ if and only if its leading term with respect to $>_{-1}$ is a multiple of $\mathbf{e}_2$. The elements of K given by Theorem (4.8) have this property and have minimal possible $\deg(\Lambda)$, so their leading term is minimal among leading terms which are multiples of $\mathbf{e}_2$.

Aiming for a contradiction now, suppose that $\overline{g}$ is not minimal, or equivalently that there is some nonzero $h = (A, B)$ in K such that $\mathrm{LT}(h) <_{-1} \mathrm{LT}(\overline{g})$. Then by the remarks above, $\mathrm{LT}(h)$ must be a multiple of $\mathbf{e}_1$, that is, it must appear in A, so

$$\deg(\overline{\Lambda}) > \deg(A) \geq \deg(B). \tag{4.17}$$

But both h and $\overline{g}$ are solutions of the key equation:

$$A \equiv SB \bmod x^{2t}$$
$$\overline{\Omega} \equiv S\overline{\Lambda} \bmod x^{2t}.$$

Multiplying the second congruence by B, the first by $\overline{\Lambda}$, and subtracting, we obtain

$$\overline{\Lambda}A \equiv B\overline{\Omega} \bmod x^{2t}. \tag{4.18}$$

We claim this contradicts the inequalities on degrees above. Recall that $\deg(\overline{\Lambda}) \leq t$ and $\deg(\overline{\Omega}) < \deg(\overline{\Lambda})$, hence $\deg(\overline{\Omega}) \leq t - 1$. But from (4.17), it follows that $\deg(A) \leq t - 1$. The product on the left of (4.18) has degree at most $2t - 1$, and the product on the right side has degree strictly less than the product on the left. But that is absurd. □

Combining (4.16) and (4.15), we see that the special solution of the key equation that we seek can be found in a Gröbner basis for K with respect to the $>_{-1}$ order. This gives at least *two* possible ways to proceed in decoding.

1. We could use the generating set
$$\{(S, 1), (x^{2t}, 0)\}$$
for K, apply Buchberger's algorithm (or a suitable variant adapted to the special properties of modules over the one variable polynomial ring $\mathbb{F}_q[x]$), and compute a Gröbner basis for K with respect to $>_{-1}$ directly. Then the minimal element $\overline{g}$ which solves the decoding problem will appear in the Gröbner basis.
2. Alternatively, we could make use of the fact recorded in Corollary (4.13). Since $\mathcal{G} = \{(S, 1), (x^{2t}, 0)\}$ is already a Gröbner basis for K with respect to another order, and $\mathbb{F}_q[x]^2/M$ is finite-dimensional over $\mathbb{F}_q$, we can use an extension of the FGLM Gröbner basis conversion algorithm from §3 of Chapter 2 (see [Fit2]) to convert $\{(S, 1), (x^{2t}, 0)\}$ into a Gröbner basis

$\mathcal{G}'$ for the same module, but with respect to the $>_{-1}$ order. Then as in approach 1, the minimal element in K will be an element of $\mathcal{G}'$.

Yet another possibility would be to build up to the desired solution of the key equation inductively, solving the congruences

$$\Omega \equiv \Lambda S \bmod x^{\ell}$$

for $\ell = 1, 2, \ldots, 2t$ in turn. This approach gives one way to understand the operations from the Berlekamp-Massey algorithm mentioned above. See [Fit1] for a Gröbner basis interpretation of this method.

Of the two approaches detailed above, a deeper analysis shows that the first approach is more efficient for long codes. But both are interesting from the mathematical standpoint. We will discuss the second approach in the text to conclude this section, and indicate how the first might proceed in the exercises. One observation we can make here is that the *full* analog of the FGLM algorithm need not be carried out. Instead, we need only consider the monomials in $\mathbb{F}_q[x]^2$ one by one in increasing $>_{-1}$ order and stop on the *first* instance of a linear dependence among the remainders of those monomials on division by $\mathcal{G}$. Here is the algorithm (see [Fit2], Algorithm 3.5). It uses a subalgorithm called *nextmonom* which takes a monomial u and returns the next monomial after u in $\mathbb{F}_q[x]^2$ in the $>_{-1}$ order. (Since we will stop after one element of the new Gröbner basis is obtained, we do not need to check whether the next monomial is a multiple of the leading terms of the other new basis elements as we did in the full FGLM algorithm in Chapter 2.)

(4.19) Proposition. *The following algorithm computes the minimal element of the module K of solutions of the key equation with respect to the $>_{-1}$ order.*

Input: $\mathcal{G} = \{(S, 1), (x^{2t}, 0)\}$
Output: $(\overline{\Omega}, \overline{\Lambda})$ *minimal in* $K = \langle \mathcal{G} \rangle$ *with respect to* $>_{-1}$
Uses: Division algorithm with respect to $\mathcal{G}$, *using* $>_{\deg(S)}$ *order*, *nextmonom*

$t_1 := (0, 1)$; $R_1 := \overline{t_1}^{\mathcal{G}}$
done := false
WHILE done = false DO
  $t_{j+1} := nextmonom(t_j)$
  $R_{j+1} := \overline{t_{j+1}}^{\mathcal{G}}$
  IF there are $c_i \in \mathbb{F}_q$ *with* $R_{j+1} = \sum_{i=1}^{j} c_i R_i$ *THEN*

$$(\overline{\Omega}, \overline{\Lambda}) := t_{j+1} - \sum_{i=1}^{j} c_i t_i$$

done := true

ELSE

$j := j + 1$

Exercise 5. Prove that this algorithm always terminates and correctly computes the minimal element of $K = \langle \mathcal{G} \rangle$ with respect to $>_{-1}$. Hint: See the proof of Theorem (3.4) in Chapter 2; this situation is simpler in several ways, though.

We illustrate the decoding method based on this algorithm with an example. Let C be the Reed-Solomon code over $\mathbb{F}_9$, with

$$g = (x - \alpha)(x - \alpha^2)(x - \alpha^3)(x - \alpha^4),$$

and $d = 5$. We expect to be able to correct any error vector of weight 2 or less. We claim that

$$c = x^7 + 2x^5 + x^2 + 2x + 1$$

is a codeword for C. This follows for instance from a Maple computation such as this one. After initializing the field (`a` below is the primitive element α for $\mathbb{F}_9$), setting c equal to the polynomial above, and g equal to the generator,

```
Rem(c,g,x) mod 3;
```

returns 0, showing that g divides c.

Suppose that errors occur in transmission of c, yielding the received word

$$y = x^7 + \alpha x^5 + (\alpha + 2)x^2 + 2x + 1.$$

(Do you see where the errors occurred?) We begin by computing the syndrome S. Using Maple, we find $y(\alpha) = \alpha + 2$, $y(\alpha^2) = y(\alpha^3) = 2$, and $y(\alpha^4) = 0$. For example, the calculation of $y(\alpha)$ can be done simply by initializing the field, defining y as above, then computing

```
Normal(subs(x=a,y)) mod 3;
```

So we have

$$S = 2x^2 + 2x + \alpha + 2.$$

By Theorem (4.8), we need to consider the module K of solutions of the key equation

$$\Omega \equiv \Lambda S \bmod x^4.$$

By Corollary (4.13), $\mathcal{G} = \{(x^4, 0), (2x^2 + 2x + \alpha + 2, 1)\}$ is the reduced Gröbner basis for K with respect to the order $>_2$. Applying Proposition (4.19), we find

$$\begin{aligned} t_1 &= (0,1) & R_1 &= (x^2 + x + 2\alpha + 1, 0) \\ t_2 &= (1,0) & R_2 &= (1,0) \\ t_3 &= (0,x) & R_3 &= (x^3 + x^2 + (2\alpha+1)x, 0) \\ t_4 &= (x,0) & R_4 &= (x,0) \\ t_5 &= (0,x^2) & R_5 &= (x^3 + (2\alpha+1)x^2, 0). \end{aligned}$$

Here for the first time we obtain a linear dependence:

$$R_5 = -(\alpha R_1 + (\alpha+1)R_2 + 2R_3 + (\alpha+1)R_4).$$

Hence,

$$\alpha t_1 + (\alpha+1)t_2 + 2t_3 + (\alpha+1)t_4 + t_5 = (\alpha + 1 + (\alpha+1)x, \alpha + 2x + x^2)$$

is the minimal element $(\overline{\Omega}, \overline{\Lambda})$ of K that we are looking for.

The error locations are found by solving

$$\overline{\Lambda(x)} = x^2 + 2x + \alpha = 0.$$

Recall, by definition $\Lambda = \prod_{i \in I}(1 - \alpha^i x)$ has constant term 1, so we need to adjust constants to get the actual error locator polynomial and the correct Ω to use in the determination of the error values, using the Forney formula of Exercise 1. Dividing by α, we obtain $\Lambda = (\alpha+1)x^2 + (2\alpha+2)x + 1$. By factoring, or by an exhaustive search for the roots as in

```
for j to 8 do
  Normal(subs(x = a^j,Lambda) mod 3;
end do;
```

we find that the roots are $x = \alpha^3$ and $x = \alpha^6$. Taking the exponents of the *inverses* gives the error locations: $(\alpha^3)^{-1} = \alpha^5$ and $(\alpha^6)^{-1} = \alpha^2$, so the errors occurred in the coefficients of x^2 and x^5. (Check the codeword c and the received word y above to see that this is correct.) Next, we apply Exercise 1 to obtain the error values. We have

$$\Omega = (1/\alpha)((\alpha+1)x + \alpha + 1) = (\alpha+2)x + \alpha + 2.$$

For the error location $i = 2$, for instance, we have $\chi_2(x) = 1 - \alpha^5 x$, and

$$\begin{aligned} e_2 &= \frac{\Omega(\alpha^{-2})}{\alpha^2 \chi_2(\alpha^{-2})} \\ &= \alpha + 1. \end{aligned}$$

This also checks. The error value $e_5 = \alpha + 1$ is determined similarly; to decode we subtract $e = (\alpha+1)x^5 + (\alpha+1)x^2$ from y, and we recover the correct codeword.

In the Exercises below, we will consider how (part of) a direct calculation of the Gröbner basis for K with respect to $>_{-1}$ can also be used for decoding. Other applications of computational commutative algebra to coding theory are discussed in [dBP1] and [dBP2].

Additional Exercises for §4

Exercise 6. Let $(\overline{\Omega}, \overline{\Lambda})$ be any solution of the congruence (4.6), where S is the syndrome polynomial for some correctable error.

a. Writing $\overline{\Lambda} = \sum_{i=0}^{t} \Lambda_i x^i$ and $S = \sum_{j=1}^{2t} E_j x^{j-1}$ show that (4.6) yields the following system of t homogeneous linear equations for the $t+1$ coefficients in $\overline{\Lambda}$:

$$\sum_{k=0}^{t} \Lambda_k E_{t+\ell-k} = 0 \tag{4.20}$$

for each $\ell = 1, \ldots, t$.

b. Assuming no more than t errors occurred, say in the locations given by a set of indices I, $E_{t+\ell-k} = \sum_{i \in I} e_i \alpha^{i(t+\ell-k)}$ for some polynomial $e(x)$ with t or fewer nonzero terms. Substitute in (4.20) and rearrange to obtain

$$\begin{aligned} 0 &= \sum_{k=0}^{t} \Lambda_k E_{t+\ell-k} \\ &= \sum_{i \in I} e_i \overline{\Lambda}(\alpha^{-i}) \alpha^{i(t+\ell)}. \end{aligned} \tag{4.21}$$

c. Show that the last equation in (4.21) implies that $\overline{\Lambda}(\alpha^{-i}) = 0$ for all $i \in I$, which gives another proof that Λ divides $\overline{\Lambda}$. Hint: The equations in (4.21) can be viewed as a system of homogeneous linear equations in the unknowns $e_i \overline{\Lambda}(\alpha^{-i})$. The matrix of coefficients has a notable special form. Also, $e_i \neq 0$ for $i \in I$.

Solving the decoding problem can be rephrased as finding the linear recurrence relation (4.20) of minimal order for the E_j sequence. The coefficients Λ_k then give the error locator polynomial.

Exercise 7. A *direct analog* of syndrome decoding for Reed-Solomon codes might begin by computing the remainder on division of a received word y by the generator, giving an expression $y = c + R$, where c is a codeword. How is the remainder R related to the error polynomial e? Is this c necessarily the nearest codeword to y? (There is another decoding method for Reed-Solomon codes, due to Welch and Berlekamp, that uses R rather than the syndrome S. It can also be rephrased as solving a key equation, and Gröbner bases can be applied to solve that equation also.)

Exercise 8. Prove Corollary (4.13).

Exercise 9. Prove Proposition (4.15). Hint: Think about the definition of a Gröbner basis.

Exercise 10. Consider the Reed-Solomon code over $\mathbb{F}_9$ with generator polynomial $g = (x - \alpha)(x - \alpha^2)$ ($d = 3$, so this code is 1 error-correcting). Perform computations using Proposition (4.19) to decode the received words

$$y(x) = x^7 + \alpha x^5 + (\alpha + 2)x^3 + (\alpha + 1)x^2 + x + 2,$$

and

$$y(x) = x^7 + x^6 + \alpha x^5 + (\alpha + 1)x^3 + (\alpha + 1)x^2 + x + 2\alpha.$$

What are the solutions of $\Lambda = 0$ in the second case? How should the decoder handle the situation?

Exercise 11. In this and the following exercise, we will discuss how a portion of a direct calculation of the Gröbner basis for K with respect to $>_{-1}$ starting from the generating set $\{g_1, g_2\} = \{(x^{2t}, 0), (S, 1)\}$ can also be used for decoding. Consider the first steps of Buchberger's algorithm. Recall that S has degree $2t - 1$ or less.

a. Show that the first steps of the algorithm amount to applying the 1-variable division algorithm to divide S into x^{2t}, yielding an equation $x^{2t} = qS + R$, with a quotient q of degree 1 or more, and a remainder R that is either 0 or of degree smaller than $\deg S$. This gives the equation

$$(x^{2t}, 0) = q(S, 1) + (R, -q).$$

b. Deduce that g_2 and $g_3 = (R, -q)$ also generate the module K, so g_1 can actually be discarded for the Gröbner basis computation.

c. Proceed as in the Euclidean algorithm for polynomial GCD's (see e.g. [CLO], Chapter 1, §5), working on the *first* components. For instance, at the next stage we find a relation of the form

$$(S, 1) = q_1(R, -q) + (R_1, q_1 q + 1).$$

In the new module element, $g_4 = (R_1, q_1 q + 1)$, the degree of the first component has decreased, and the degree of the second has increased. Show that after a finite number of steps of this process, we will produce an element (Ω, Λ) of the module K whose second component has degree greater than the degree of the first, so that its $>_{-1}$ leading term is a multiple of $\mathbf{e}_2$.

d. Show that the element obtained in this way is a minimal element K with respect to $>_{-1}$. Hint: It is easy to see that a minimal element could be obtained by removing any factors common to the two components of

this module element; by examining the triple $(\Omega, \Lambda, \Gamma)$ obtained as a solution of the explicit form of the key equation: $\Omega = \Lambda S + x^{2t}\Gamma$, show that in fact Ω and Λ are automatically relatively prime.

Exercise 12. Apply the method from Exercise 11 to the decoding problem from the end of the text of this section. Compare your results with those of the other method. Also compare the amount of calculation needed to carry out each one. Is there a clear "winner"?

Exercise 13. Apply the method from Exercise 11 to the decoding problems from Exercise 10.