Galois fields

6

We have seen that the mathematical framework for linear codes and cyclic codes are matrices and polynomials respectively. For each type of code, encoding and decoding can be concisely formulated in terms of their respective mathematics. This difference in mathematics signifies more than just mathematical representation, for in going from linear codes to cyclic codes there is an increase in underlying mathematical structure. Moving on, from cyclic codes to the next level of codes with greater mathematical structure are the BCH codes, considered in Chapter 7. The mathematical framework within which the BCH codes are addressed is that of Galois fields and it is these that we consider next.

6.1 Roots of equations

We have already considered fields, in particular finite fields, in Chapter 5. Here we are interested in finite fields constructed from the roots of equations. The motivation for this approach lies in the property of cyclic codes that all codeword polynomials $c(x)$ have the generator polynomial $g(x)$ as a factor. This can be restated by saying that any root of $g(x)=0$ is also a root of $c(x)=0$ and it is the exploitation of roots of equations that takes us to the BCH codes and to the finite fields relevant to the codes.

Consider the *algebraic equation* of the form

$$a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 = 0 \tag{6.1}$$

where $a_0 \neq 0$. For now the coefficients $a_0, a_1, \ldots, a_n$ are taken to be real numbers but later we consider eqn 6.1 when the coefficients are binary. The simplest algebraic equation is the equation of first degree obtained by setting $n=1$, this is usually written as

$$ax + b = 0$$

where $a = a_1$, $b = a_0$ and there exists one root $x = -b/a$. The second degree (quadratic) equation, commonly expressed as

$$ax^2 + bx + c = 0$$

where now $a = a_2$, $b = a_1$ and $c = a_0$, has 2 roots given by the famous 'formula' for solving quadratic equations

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{6.2}$$

The roots of equations of third and fourth degree, with 3 and 4 roots respectively, can likewise be obtained algebraically. However, the same is not true for fifth-degree equations for it has been shown that it is not possible to solve the fifth-degree equation using a finite number of algebraic operations, in other words it is impossible to obtain the roots algebraically. This was established in the early 19th century by Abel, a Norwegian mathematician. Furthermore, it is now known that algebraic equations of degree greater than 5 cannot be solved algebraically. This though does not mean that roots for fifth and higher degree equations do not exist, it is just that expressions for giving the roots do not exist. A fifth-degree equation has 5 roots and an equation with degree n, as given by eqn 6.1, has n roots. There is, however, one slight problem regarding the nature of some of the roots. Returning to the quadratic equation $ax^2 + bx + c = 0$ and setting $a = c = 1$ and $b = 0$ gives

$$x^2 + 1 = 0. \tag{6.3}$$

Each root of eqn 6.3 has the property that when multiplied by itself it gives -1. But there are no real numbers that have this property and the only way of obtaining 2 roots for eqn 6.3 is by defining a new *imaginary* or *complex* number j, which satisfies

$$j^2 + 1 = 0. \tag{6.4}$$

The number j can be combined with 2 real numbers p and q to give the complex number $p + jq$. Addition and multiplication over the set of all complex numbers obey the rules described in Section 5.3 for a set of elements to form a field, and the resulting field is known as the *complex field*. It is because of the existence of the complex field that all equations of degree n have n roots. The roots to eqn 6.1 are of the form $p + jq$ with real roots having $q = 0$. If we consider a quadratic equation with $b \neq 0$, for example

$$x^2 - 4x + 13 = 0$$

then using eqn 6.2 we can easily find that there are two complex roots $2 - j3$ and $2 + j3$. Complex roots, of an equation with real coefficients, always occur in pairs known as *complex conjugates* or *conjugates*, which take the form $p \pm jq$. The complex number $p - jq$ is the conjugate of $p + jq$, and likewise $p + jq$ is the conjugate of $p - jq$. An equation can never have an odd number of complex roots, this would require a complex root without a conjugate. In a cubic equation the roots are either all real or there is one real root and two roots that are complex conjugates. For example, the roots of

$$x^3 - 6x^2 + 13x - 20 = 0$$

are 4, $1 + j2$ and $1 - j2$.

The complex field includes all the real numbers as they can be considered as field elements with $q = 0$. We can think of the complex field as expanding or extending the real field that we started with. In this sense, the real field is referred to as a *base field* and the complex field as an *extension field*, the complex field is an extension of the real field. The coefficients $a_0, a_1, \ldots, a_n$ in eqn 6.1 belong to the base field (i.e. the real field) whereas the roots belong to the extension field (i.e. the complex field). The occurrence of complex roots as conjugate pairs is necessary for the coefficients

to lie in the real field. To illustrate this, let's try to construct a quadratic equation with 2 roots that are not conjugate pairs. For example let $1 + j2$ and $3 - j4$ be roots of a quadratic equation, then the quadratic equation will be

$$(x - (1 + j2))(x - (3 - j4)) = x^2 - (4 - j2)x + (11 + j2)$$

which has coefficients in the complex field. It can be easily shown that if the coefficients of a quadratic equation, with complex roots, lie in the real field then the roots must form a conjugate pair.

Example 6.1
If $p + jq$ is a root of a quadratic equation with real coefficients, show that its conjugate $p - jq$ is the other root.

Let $a + jb$ be the other root, then the quadratic equation is

$$(x - (p + jq))(x - (a + jb)) = 0$$

which gives

$$x^2 - x[(a + p) + j(b + q)] + (pa - qb) + j(aq + bp) = 0.$$

If the quadratic equation is to have real coefficients then the complex terms must equal zero, and so

$$b + q = 0$$
$$aq + bp = 0$$

which give $b = -q$ and $a = p$. Hence $a + jb = p - jq$ and so the two roots are conjugates of each other. □

We next consider roots of equations of the form $p(x) = 0$ where $p(x)$ is a polynomial with binary coefficients. As we have seen such polynomials are used for encoding and decoding binary cyclic codes. The trivial cases of $x + 1 = 0$ and $x^2 + 1 = 0$ have 1 as a root, in the latter case 1 is a double root. The quadratic equation

$$x^2 + x + 1 = 0 \tag{6.5}$$

presents more of a problem. We cannot use eqn 6.2 for solving this because eqn 6.2 has a 2 on the denominator and $2 = 0$ when using modulo-2 arithmetic. Since x can only have a value of 0 or 1, we can substitute 0 and 1 directly into eqn 6.5 to see which, if any, is a root. Substituting $x = 0$ into eqn 6.5 gives 1 and so 0 is not a root. Likewise $x = 1$ gives 1 and therefore neither 0 or 1 are roots of eqn 6.5. As another example consider

$$x^3 + x + 1 = 0. \tag{6.6}$$

Substituting $x = 0$ or $x = 1$ into this gives 1, and so again we have a binary polynomial without any binary roots. This is analogous to the situation encountered previously where we considered a real quadratic equation without any real roots, so the complex term j is 'invented' to get around the problem. Here we proceed in the same manner by defining a new term such that it is a root of the polynomial of interest. We will consider eqn 6.6 instead of eqn 6.5 as this proves to be more interesting. We could use j to denote a root of eqn 6.6, but this may cause confusion with the use of j in ordinary complex numbers. Instead it is conventional to use α to represent the newly defined root. Substituting $x = \alpha$ into eqn 6.6 gives

$$\alpha^3 + \alpha + 1 = 0. \tag{6.7}$$

Whilst eqn 6.7 may define the new root α it tells us little else about it and furthermore there are 2 more roots of eqn 6.6 that we need to find. We know that α does not belong to the binary field and to proceed further we need to determine the mathematical structure of the field within which α lies. The root α lies within a finite field known as $GF(2^3)$ which can be generated from eqn 6.7. Once $GF(2^3)$ has been established the other roots can be found.

6.2 The Galois field $GF(2^3)$

The field $GF(2^3)$ can be generated from the newly defined element α given by eqn 6.7. First consider addition and multiplication of α with the binary numbers 0 and 1. The binary numbers 0 and 1 form additive and multiplicative identity elements respectively, so

$$\alpha + 0 = \alpha$$
$$\alpha 1 = \alpha.$$

The additive inverse of α is α itself, as can be easily shown.

$$\alpha + \alpha = 1\alpha + 1\alpha = (1 + 1)\alpha = 0\alpha = 0$$

and so

$$\alpha + \alpha = 0.$$

Furthermore rearranging this gives

$$\alpha = -\alpha$$

and therefore subtraction and addition of α are equivalent. The multiplicative inverse of α is defined as

$$\alpha^{-1} = \frac{1}{\alpha}$$

so that

$$\alpha^{-1}\alpha = \frac{1}{\alpha}\alpha = 1.$$

Table 6.1 summarizes the identity and inverse elements of α.

In eqn 6.7 it is implicit that $\alpha^3 = \alpha\alpha\alpha$ and likewise other powers of α can be defined, for example $\alpha^2 = \alpha\alpha$. Higher powers of α can be determined by rearranging

Table 6.1
Identity and inverse elements of α

Identity elements	Additive	0,	$\alpha + 0 = \alpha$
	Multiplicative	1,	$\alpha 1 = \alpha$
Inverse elements	Additive	α,	$\alpha + \alpha = 0$
	Multiplicative	α^{-1}	$\alpha\alpha^{-1} = 1$

eqn 6.7 to give $\alpha^3 = \alpha + 1$ (recall that $-\alpha = \alpha$), repeatedly multiplying by α and substituting $\alpha + 1$ for α^3 whenever α^3 appears. Starting first with α^3

$$\begin{aligned}
\alpha^3 &= \alpha + 1 \\
\alpha^4 &= \alpha\alpha^3 = \alpha(\alpha + 1) = \alpha^2 + \alpha \\
\alpha^5 &= \alpha\alpha^4 = \alpha(\alpha^2 + \alpha) = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1 \\
\alpha^6 &= \alpha\alpha^5 = \alpha(\alpha^2 + \alpha + 1) = \alpha^3 + \alpha^2 + \alpha = \alpha^2 + 1.
\end{aligned} \tag{6.8}$$

The four elements α^3, α^4, α^5, and α^6 differ from each other and from the four elements 0, 1, α, and α^2. Equations 6.8 are referred to as the *polynomial representations* of the elements α^3, α^4, α^5, and α^6. It may appear that other elements can be constructed by taking further powers of α. This though is not so, for the next power of α gives

$$\alpha^7 = \alpha\alpha^6 = \alpha(\alpha^2 + 1) = \alpha^3 + \alpha = \alpha + 1 + \alpha = 1$$

and therefore

$$\alpha^7 = 1$$

which is an existing element. Forming further powers of α always generates one of the existing nonzero elements. For example, the next three powers give

$$\begin{aligned}
\alpha^8 &= \alpha\alpha^7 = \alpha 1 = \alpha \\
\alpha^9 &= \alpha\alpha^8 = \alpha\alpha = \alpha^2 \\
\alpha^{10} &= \alpha\alpha^9 = \alpha\alpha^2 = \alpha^3.
\end{aligned}$$

A field element with power greater than 6 can be reduced to an element with power of 6 or less by removing factors of α^7. This is equivalent to taking the power modulo-7. For example

$$\begin{aligned}
\alpha^{12} &= \alpha^7\alpha^5 = \alpha^5 (12 = 5 \text{ modulo-7}) \\
\alpha^{17} &= \alpha^7\alpha^7\alpha^3 = \alpha^3 (17 = 3 \text{ modulo-7})
\end{aligned}$$

and so forth. Taking into account 0 and 1 we see that we have constructed a set with the 8 elements

$$0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \text{ and } \alpha^6.$$

Along with the operations addition and multiplication the set forms a field, namely the field $GF(2^3)$. Unlike the field of real numbers, which has an infinite number of field elements, $GF(2^3)$ has a finite number of field elements and is therefore a finite field. Finite fields are also referred to as *Galois fields* after the mathematician Evariste Galois (1811–1832). The fields are usually expressed as $GF(p^m)$ where p is the number of elements in the base field, which is referred to as the field's *characteristic*, and m is the degree of the polynomial whose root is used to construct the field. The order of the field is given by $q = p^m$.

Table 6.2 shows the 8 elements of $GF(2^3)$ along with the polynomial representations of α^3, α^4, α^5, and α^6 in terms of 1, α, and α^2. We have already seen that $\alpha + \alpha = 0$, likewise any field element added to itself gives zero. For example

$$\alpha^3 + \alpha^3 = 1\alpha^3 + 1\alpha^3 = \alpha^3(1 + 1) = 0.$$

Two different field elements can be added together by using their polynomial representations. For example, adding α^4 and α^6 gives

$$\alpha^4 + \alpha^6 = (\alpha^2 + \alpha) + (\alpha^2 + 1) = \alpha + 1 = \alpha^3.$$

Example 6.2
Find (a) $\alpha^2 + \alpha$, (b) $\alpha^5 + \alpha + 1$, and (c) $\alpha^6 + \alpha^2 + 1$.

(a) From Table 6.2 we see that $\alpha^2 + \alpha = \alpha^4$.

(b) Here we first rewrite α^5 in terms of its polynomial representation and then cancel out equal field elements

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

(c) We can express α^6 as $\alpha^2 + 1$, and so

$$\alpha^6 + \alpha^2 + 1 = (\alpha^2 + 1) + \alpha^2 + 1 = 0.$$

□

Using the polynomial representation of field elements to add 2 elements together is rather tedious. It is easier to construct a table showing the results of all additions and then to refer to the table when needed. Table 6.3(a) shows addition of elements within $GF(2^3)$.

Table 6.2
The field elements of $GF(2^3)$

0
1
α
α^2
$\alpha^3 = \alpha + 1$
$\alpha^4 = \alpha^2 + \alpha$
$\alpha^5 = \alpha^2 + \alpha + 1$
$\alpha^6 = \alpha^2 + 1$

Table 6.3
Addition and multiplication in $GF(2^3)$

(a) *Addition*

+	0	1	α	α^2	α^3	α^4	α^5	α^6
0	0	1	α	α^2	α^3	α^4	α^5	α^6
1	1	0	α^3	α^6	α	α^5	α^4	α^2
α	α	α^3	0	α^4	1	α^2	α^6	α^5
α^2	α^2	α^6	α^4	0	α^5	α	α^3	1
α^3	α^3	α	1	α^5	0	α^6	α^2	α^4
α^4	α^4	α^5	α^2	α	α^6	0	1	α^3
α^5	α^5	α^4	α^6	α^3	α^2	1	0	α
α^6	α^6	α^2	α^5	1	α^4	α^3	α	0

(b) *Multiplication*

×	0	1	α	α^2	α^3	α^4	α^5	α^6
0	0	0	0	0	0	0	0	0
1	0	1	α	α^2	α^3	α^4	α^5	α^6
α	0	α	α^2	α^3	α^4	α^5	α^6	1
α^2	0	α^2	α^3	α^4	α^5	α^6	1	α
α^3	0	α^3	α^4	α^5	α^6	1	α	α^2
α^4	0	α^4	α^5	α^6	1	α	α^2	α^3
α^5	0	α^5	α^6	1	α	α^2	α^3	α^4
α^6	0	α^6	1	α	α^2	α^3	α^4	α^5

Multiplying two field elements together is straightforward, their powers are added together and because $\alpha^7 = 1$ factors of α^7 can be removed. For example, the product of α^3 and α^6 gives

$$\alpha^3\alpha^6 = \alpha^{3+6} = \alpha^9 = \alpha^7\alpha^2 = 1\alpha^2 = \alpha^2.$$

This is the same as taking the sum modulo-7 of the two powers. In the above case we have $3 + 6 = 2$ modulo-7 and the resulting field element is α^2 as obtained. Therefore the product of the field elements α^i and α^j in $GF(2^3)$ is

$$\alpha^i\alpha^j = \alpha^{(i+j)\text{modulo-7}}$$

Although there is no need to refer to a multiplication table when multiplying field elements, one is included for completeness (see Table 6.3b).

Example 6.3
Find (a) $\alpha\alpha^5$, (b) $\alpha^4\alpha^5$, and (c) $\alpha^5\alpha^6\alpha^4$.

(a) $\alpha\alpha^5 = \alpha^6$
(b) $\alpha^4\alpha^5 = \alpha^9 = \alpha^7\alpha^2 = \alpha^2$
(c) $\alpha^5\alpha^6\alpha^4 = \alpha^{15} = \alpha^7\alpha^7\alpha = \alpha$. □

For any field element α^i in $GF(2^3)$ we have

$$\alpha^i + \alpha^i = 1\alpha^i + 1\alpha^i = \alpha^i(1 + 1) = 0$$

and therefore each element is its own additive inverse. The multiplicative inverse of α^i is defined as the element α^{-i} such that

$$\alpha^i\alpha^{-i} = 1$$

and is given by

$$\boxed{\alpha^{-i} = \alpha^{7-i}.}$$

Taking the product of α^i with α^{-i} gives $\alpha^i\alpha^{-i} = \alpha^{i+7-i} = \alpha^7 = 1$ and so α^{-i} satisfies the requirement for a multiplicative inverse. Take, for instance, the multiplicative inverse of α^3

$$\alpha^{-3} = \alpha^{7-3} = \alpha^4$$

and so the multiplicative inverse of α^3 is α^4. Note that the element 1 is its own multiplicative inverse.

One of the requirements of a field is that division by nonzero elements is possible. Given α^i and α^j in $GF(2^3)$, where $\alpha^j \neq 0$, then α^i divided by α^j is

$$\frac{\alpha^i}{\alpha^j} = \alpha^i\alpha^{-j} = \alpha^{(i-j)\text{modulo-7}}$$

where α^{-j} is the multiplicative inverse of α^j. Note that if $i - j < 0$ then $(i - j)$ modulo-7 is found by adding 7 to $i - j$. Also if $i = j$ then clearly $\alpha^i/\alpha^j = 1$. For example

$$\frac{\alpha^6}{\alpha^2} = \alpha^6\alpha^{-2} = \alpha^{(6-2)\text{modulo-7}} = \alpha^4.$$

The above calculation can also be thought of as follows

$$\frac{\alpha^6}{\alpha^2} = \alpha^6\alpha^{-2} = \alpha^6\alpha^5 = \alpha^{11\text{modulo-7}} = \alpha^4$$

where we have now made use of α^5 the multiplicative inverse of α^2, either way the same answer is obtained. Consider α divided by α^5, if we use the multiplicative inverse of α^5, which is α^2, we get

$$\frac{\alpha}{\alpha^5} = \alpha\alpha^{-5} = \alpha\alpha^2 = \alpha^3$$

or without using the inverse we can view the calculation as

$$\frac{\alpha}{\alpha^5} = \alpha\alpha^{-5} = \alpha^{(1-5)\text{modulo-7}} = \alpha^{(-4)\text{modulo-7}} = \alpha^3.$$

Example 6.4
Find (a) α^2/α^5, (b) $1/\alpha$, (c) α^3/α, and (d) α/α^3.

(a) The inverse of α^5 is α^2, and so $\alpha^2/\alpha^5 = \alpha^2\alpha^2 = \alpha^4$. Or we can view this as $\alpha^2/\alpha^5 = \alpha^{(2-5)\text{modulo-7}} = \alpha^4$.
(b) $1/\alpha = \alpha^6$
(c) $\alpha^3/\alpha = \alpha^2$
(d) $\alpha/\alpha^3 = \alpha^5$. □

We now return to the problem of finding the three roots of the binary equation $x^3 + x + 1 = 0$ (see Section 6.1). We have already 'found' 1 root, namely α belonging to $GF(2^3)$, and next we will use a trial-and-error method to test the other elements of $GF(2^3)$ to see if they satisfy $x^3 + x + 1 = 0$. Only 5 of the 8 elements need to be

considered as α is a root by definition and 0 and 1 are known not to be roots. Starting with $x = \alpha^2$ gives

$$\alpha^6 + \alpha^2 + 1 = 1 + 1 = 0$$

and so α^2 is a root (here we have referred to Table 6.3(a) to get $\alpha^2 + \alpha^6 = 1$). Next try $x = \alpha^3$

$$\alpha^9 + \alpha^3 + 1 = \alpha^3\alpha^2 + \alpha^3 + 1 = \alpha^2 + \alpha = \alpha^4 \neq 0$$

and so α^3 is not a root. Continuing with $x = \alpha^4$ gives

$$\alpha^{12} + \alpha^4 + 1 = \alpha^5 + \alpha^4 + 1 = 1 + 1 = 0$$

and therefore α^4 is the third root. The elements α^5 and α^6 cannot be roots because a cubic equation can only have 3 roots. As a check, substituting α^5 and α^6 into $x^3 + x + 1$ gives α^2 and α respectively, thus confirming that they are not roots. Therefore the three roots of the binary equation

$$x^3 + x + 1 = 0$$

are the field elements α, α^2, and α^4 belonging to the finite field $GF(2^3)$. Hence the original aim of finding the equation's roots has been achieved.

In Section 6.1 we considered equations with real coefficients and complex roots and the idea of base and extension fields were introduced in the context of the real and complex fields. The real field is thought of as the base field containing the equations' coefficients. The extension field contains the base field and extends it to include the complex roots. In the present case the equation of interest $x^3 + x + 1 = 0$ has its coefficients in the binary field and its roots in $GF(2^3)$. The binary field is denoted by **GF(2)** as it is a finite field with 2 elements. The field $GF(2^3)$ is an extension field of the binary field $GF(2)$. Note that $GF(2^3)$ is not the only extension field of $GF(2)$. The field $GF(2^3)$ has been constructed by determining the roots of the polynomial $p(x) = x^3 + x + 1$. Other extension fields can be generated using different polynomials. However, not all polynomials can generate extension fields, this is considered further in Section 6.5.

6.3 The fields $GF(2^4)$ and $GF(2^5)$

Here we are going to first construct the finite field $GF(2^4)$ and then take a brief look at $GF(2^5)$. $GF(2^4)$ is a field that the reader will encounter in most text books on error control. $GF(2^3)$ was constructed using a cubic polynomial that does not have 0 or 1 as roots. Consider the polynomial

$$p(x) = x^4 + x + 1. \tag{6.9}$$

Neither 0 or 1 are roots of $p(x) = 0$, it can be easily seen that $p(0) = p(1) = 1$. The four roots of eqn 6.9 therefore lie outside the binary field $GF(2)$. If we let α be one of the roots, then $p(\alpha) = 0$ by definition and

$$\alpha^4 + \alpha + 1 = 0 \tag{6.10}$$

Equation 6.10 is used to generate $GF(2^4)$ in the same way as eqn 6.7 was used to generate $GF(2^3)$. The binary elements 0 and 1 are again additive and multiplicative identity elements of α respectively. To determine the elements of $GF(2^4)$ we proceed in the same manner as when constructing $GF(2^3)$, by forming successive powers of α until an existing element is generated. Rearranging 6.10 gives

$$\alpha^4 = \alpha + 1 \tag{6.11}$$

When constructing higher powers of α eqn 6.11 is used to reduce field elements to their lowest power. Starting with α^4 and successively multiplying by α gives

$$\begin{aligned}
\alpha^4 &= \alpha + 1 \\
\alpha^5 &= \alpha\alpha^4 = \alpha(\alpha + 1) = \alpha^2 + \alpha \\
\alpha^6 &= \alpha\alpha^5 = \alpha(\alpha^2 + \alpha) = \alpha^3 + \alpha^2 \\
\alpha^7 &= \alpha\alpha^6 = \alpha(\alpha^3 + \alpha^2) = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1
\end{aligned} \tag{6.12}$$

Note that at this point $\alpha^7 = 1$ in $GF(2^3)$. However, here α^7 differs from all the previous elements and we therefore continue producing higher powers of α until an existing element is obtained

$$\begin{aligned}
\alpha^8 &= \alpha\alpha^7 = \alpha(\alpha^3 + \alpha + 1) = \alpha^4 + \alpha^2 + \alpha = \alpha^2 + 1 \\
\alpha^9 &= \alpha\alpha^8 = \alpha(\alpha^2 + 1) = \alpha^3 + \alpha \\
\alpha^{10} &= \alpha\alpha^9 = \alpha(\alpha^3 + \alpha) = \alpha^4 + \alpha^2 = \alpha^2 + \alpha + 1 \\
\alpha^{11} &= \alpha\alpha^{10} = \alpha(\alpha^2 + \alpha + 1) = \alpha^3 + \alpha^2 + \alpha \\
\alpha^{12} &= \alpha\alpha^{11} = \alpha(\alpha^3 + \alpha^2 + \alpha) = \alpha^3 + \alpha^2 + \alpha + 1.
\end{aligned}$$

All the elements generated so far are different, so the process is continued

$$\begin{aligned}
\alpha^{13} &= \alpha\alpha^{12} = \alpha(\alpha^3 + \alpha^2 + \alpha + 1) = \alpha^3 + \alpha^2 + 1 \\
\alpha^{14} &= \alpha\alpha^{13} = \alpha(\alpha^3 + \alpha^2 + 1) = \alpha^3 + 1
\end{aligned}$$

and finally

$$\alpha^{15} = \alpha\alpha^{14} = \alpha(\alpha^3 + 1) = \alpha^4 + \alpha = 1$$

which is an existing element. Constructing further powers of α will always give existing field elements, for example

$$\begin{aligned}
\alpha^{16} &= \alpha\alpha^{15} = \alpha 1 = \alpha \\
\alpha^{17} &= \alpha\alpha^{16} = \alpha\alpha = \alpha^2.
\end{aligned}$$

The field $GF(2^4)$ therefore has the following 16 elements

$$0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7, \alpha^8, \alpha^9, \alpha^{10}, \alpha^{11}, \alpha^{12}, \alpha^{13}, \alpha^{14}.$$

Table 6.4 lists the elements along with their polynomial representations in terms of 1, α, α^2, and α^3. Field elements of $GF(2^4)$ can be added together by using the polynomial representation of elements given in Table 6.4 or by referring to an addition table (see Table 6.5a).

Table 6.4
The field elements of $GF(2^4)$

0
1
α
α^2
α^3
$\alpha^4 = \alpha + 1$
$\alpha^5 = \alpha^2 + \alpha$
$\alpha^6 = \alpha^3 + \alpha^2$
$\alpha^7 = \alpha^3 + \alpha + 1$
$\alpha^8 = \alpha^2 + 1$
$\alpha^9 = \alpha^3 + \alpha$
$\alpha^{10} = \alpha^2 + \alpha + 1$
$\alpha^{11} = \alpha^3 + \alpha^2 + \alpha$
$\alpha^{12} = \alpha^3 + \alpha^2 + \alpha + 1$
$\alpha^{13} = \alpha^3 + \alpha^2 + 1$
$\alpha^{14} = \alpha^3 + 1$

Table 6.5(a)
Addition in $GF(2^4)$

+	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	0	α^4	α^8	α^{14}	α	α^{10}	α^{13}	α^9	α^2	α^7	α^5	α^{12}	α^{11}	α^6	α^3
α	α^4	0	α^5	α^9	1	α^2	α^{11}	α^{14}	α^{10}	α^3	α^8	α^6	α^{13}	α^{12}	α^7
α^2	α^8	α^5	0	α^6	α^{10}	α	α^3	α^{12}	1	α^{11}	α^4	α^9	α^7	α^{14}	α^{13}
α^3	α^{14}	α^9	α^6	0	α^7	α^{11}	α^2	α^4	α^{13}	α	α^{12}	α^5	α^{10}	α^8	1
α^4	α	1	α^{10}	α^7	0	α^8	α^{12}	α^3	α^5	α^{14}	α^2	α^{13}	α^6	α^{11}	α^9
α^5	α^{10}	α^2	α	α^{11}	α^8	0	α^9	α^{13}	α^4	α^6	1	α^3	α^{14}	α^7	α^{12}
α^6	α^{13}	α^{11}	α^3	α^2	α^{12}	α^9	0	α^{10}	α^{14}	α^5	α^7	α	α^4	1	α^8
α^7	α^9	α^{14}	α^{12}	α^4	α^3	α^{13}	α^{10}	0	α^{11}	1	α^6	α^8	α^2	α^5	α
α^8	α^2	α^{10}	1	α^{13}	α^5	α^4	α^{14}	α^{11}	0	α^{12}	α	α^7	α^9	α^3	α^6
α^9	α^7	α^3	α^{11}	α	α^{14}	α^6	α^5	1	α^{12}	0	α^{13}	α^2	α^8	α^{10}	α^4
α^{10}	α^5	α^8	α^4	α^{12}	α^2	1	α^7	α^6	α	α^{13}	0	α^{14}	α^3	α^9	α^{11}
α^{11}	α^{12}	α^6	α^9	α^5	α^{13}	α^3	α	α^8	α^7	α^2	α^{14}	0	1	α^4	α^{10}
α^{12}	α^{11}	α^{13}	α^7	α^{10}	α^6	α^{14}	α^4	α^2	α^9	α^8	α^3	1	0	α	α^5
α^{13}	α^6	α^{12}	α^{14}	α^8	α^{11}	α^7	1	α^5	α^3	α^{10}	α^9	α^4	α	0	α^2
α^{14}	α^3	α^7	α^{13}	1	α^9	α^{12}	α^8	α	α^6	α^4	α^{11}	α^{10}	α^5	α^2	0

Example 6.5
Find (a) $\alpha^2 + \alpha^9$ and (b) $\alpha^7 + \alpha^3 + \alpha^{11}$ in $GF(2^4)$.

(a) From Table 6.4, $\alpha^9 = \alpha^3 + \alpha$ and so $\alpha^2 + \alpha^9 = \alpha^2 + \alpha^3 + \alpha = \alpha^{11}$.

(b) Again from Table 6.4, $\alpha^7 = \alpha^3 + \alpha + 1$ and $\alpha^{11} = \alpha^3 + \alpha^2 + \alpha$ and therefore

$$\begin{aligned}\alpha^7 + \alpha^3 + \alpha^{11} &= \alpha^3 + \alpha + 1 + \alpha^3 + \alpha^3 + \alpha^2 + \alpha \\ &= \alpha^3 + \alpha^2 + 1 = \alpha^{13}.\end{aligned}$$

□

When multiplying two field elements together, factors of α^{15} can be taken out and set to unity as $\alpha^{15} = 1$. This is equivalent to taking the sum modulo-15 of the exponents. Given two field elements α^i and α^j in $GF(2^4)$ their product is

$$\alpha^i \alpha^j = \alpha^{(i+j) \text{modulo-15}}.$$

Table 6.5(b) show the product of elements in $GF(2^4)$.

Example 6.6
Find (a) $\alpha^2\alpha^9$, (b) $\alpha^{13}\alpha^8$, and (c) $\alpha^7\alpha^{12}\alpha^4$ in $GF(2^4)$.

(a) $\alpha^2\alpha^9 = \alpha^{11}$

(b) $\alpha^{13}\alpha^8 = \alpha^{21} = \alpha^{15}\alpha^6 = 1\alpha^6 = \alpha^6$

(c) $\alpha^7\alpha^{12}\alpha^4 = \alpha^{(7+12+4) \text{modulo-15}} = \alpha^8$.

□

To divide two elements in $GF(2^4)$ the difference modulo-15 in the exponents is required, and so given α^i and α^j, where $\alpha^j \neq 0$, in $GF(2^4)$ then α^i divided by α^j is

$$\frac{\alpha^i}{\alpha^j} = \alpha^{(i-j) \text{modulo-15}}.$$

Table 6.5(b)
Multiplication in $GF(2^4)$

×	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
α	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1
α^2	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α
α^3	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2
α^4	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3
α^5	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4
α^6	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5
α^7	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6
α^8	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6	α^7
α^9	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8
α^{10}	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9
α^{11}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}
α^{12}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}
α^{13}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}
α^{14}	α^{14}	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}

Example 6.7
Find (a) α^{13}/α^2, (b) α^3/α^{10}, and (c) α/α^9 in $GF(2^4)$.

(a) $\alpha^{13}/\alpha^2 = \alpha^{(13-2)\text{modulo-}15} = \alpha^{11}$
(b) $\alpha^3/\alpha^{10} = \alpha^{(3-10)\text{modulo-}15} = \alpha^8$
(c) $\alpha/\alpha^9 = \alpha^{(1-9)\text{modulo-}15} = \alpha^7$. □

Each element α^i in $GF(2^4)$ is its own additive inverse and has a multiplicative inverse given by $\alpha^{-i} = \alpha^{15-i}$. Note that elements that are common to $GF(2^3)$ and $GF(2^4)$ do not have the same inverse. For example in $GF(2^4)$ the inverse of α^2 is $\alpha^{-2} = \alpha^{13}$, whereas the inverse of α^2 in $GF(2^3)$ is $\alpha^{-2} = \alpha^5$.

Returning now to the polynomial $p(x) = x^4 + x + 1$ we have established that one of its roots is α belonging to $GF(2^4)$. We can now proceed to find the other 3 roots by using a trial-and-error method in which each field element α^i of $GF(2^4)$ is tested to see if it gives $p(\alpha^i) = 0$. We have already seen that 0 and 1 are not roots of $p(x)$, and so starting with $x = \alpha^2$ we find that

$$p(\alpha^2) = 0$$
$$p(\alpha^3) = \alpha^5$$
$$p(\alpha^4) = 0.$$

So far then, three of the roots are α, α^2 and α^4, recall that the same three field elements belonging to $GF(2^3)$ are roots of $x^3 + x + 1$. Another root is required and so continuing with the search gives

$$p(\alpha^5) = 1$$
$$p(\alpha^6) = \alpha^{10}$$
$$p(\alpha^7) = \alpha^{10}$$
$$p(\alpha^8) = 0$$

and α^8 is therefore the fourth root. The remaining elements need not be tested as there can be only 4 roots, the reader may wish to verify that taking $x = \alpha^9$, α^{10}, α^{11}, α^{12}, α^{13}, and α^{14} gives $p(x) \neq 0$. The roots of the binary equation $x^4 + x + 1 = 0$ are therefore the elements α, α^2, α^4, and α^8 lying in the field $GF(2^4)$.

We now take a brief look at $GF(2^5)$ generated by

$$p(x) = x^5 + x^2 + 1.$$

Setting $p(x) = 0$ and defining an element α such that $p(\alpha) = 0$ gives

$$\alpha^5 = \alpha^2 + 1$$

which can be used to generate the field $GF(2^5)$ with 32 elements and where $\alpha^{31} = 1$ (see Table 6.6). The other 4 roots of $p(x) = x^5 + x^2 + 1 = 0$ are α^2, α^4, α^8, and α^{16} belonging to $GF(2^5)$.

The fields $GF(2^3)$, $GF(2^4)$, and $GF(2^5)$ are just 3 examples of extension fields that can be constructed from the binary field $GF(2)$. Polynomials of degrees 3, 4, and 5 were used to construct $GF(2^3)$, $GF(2^4)$, and $GF(2^5)$ respectively. To construct the

Table 6.6
The field elements of $GF(2^5)$

0	$\alpha^{15} = \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$
1	$\alpha^{16} = \alpha^4 + \alpha^3 + \alpha + 1$
α	$\alpha^{17} = \alpha^4 + \alpha + 1$
α^2	$\alpha^{18} = \alpha + 1$
α^3	$\alpha^{19} = \alpha^2 + \alpha$
α^4	$\alpha^{20} = \alpha^3 + \alpha^2$
$\alpha^5 = \alpha^2 + 1$	$\alpha^{21} = \alpha^4 + \alpha^3$
$\alpha^6 = \alpha^3 + \alpha$	$\alpha^{22} = \alpha^4 + \alpha^2 + 1$
$\alpha^7 = \alpha^4 + \alpha^2$	$\alpha^{23} = \alpha^3 + \alpha^2 + \alpha + 1$
$\alpha^8 = \alpha^3 + \alpha^2 + 1$	$\alpha^{24} = \alpha^4 + \alpha^3 + \alpha^2 + \alpha$
$\alpha^9 = \alpha^4 + \alpha^3 + \alpha$	$\alpha^{25} = \alpha^4 + \alpha^3 + 1$
$\alpha^{10} = \alpha^4 + 1$	$\alpha^{26} = \alpha^4 + \alpha^2 + \alpha + 1$
$\alpha^{11} = \alpha^2 + \alpha + 1$	$\alpha^{27} = \alpha^3 + \alpha + 1$
$\alpha^{12} = \alpha^3 + \alpha^2 + \alpha$	$\alpha^{28} = \alpha^4 + \alpha^2 + \alpha$
$\alpha^{13} = \alpha^4 + \alpha^3 + \alpha^2$	$\alpha^{29} = \alpha^3 + 1$
$\alpha^{14} = \alpha^4 + \alpha^3 + \alpha^2 + 1$	$\alpha^{30} = \alpha^4 + \alpha$

field $GF(2^m)$ a polynomial of degree m is required. In the following sections we consider some basic properties of extension fields and field elements, along with the characteristics of polynomials that are relevant to the construction of fields.

6.4 Primitive field elements

The nonzero field elements of the Galois fields are generated by taking successive multiples of a single element α. Field elements that can generate all the nonzero elements of a field are said to be *primitive* and α is primitive in $GF(2^3)$, $GF(2^4)$, and $GF(2^5)$. It can be shown that every Galois field has at least one primitive field element. In $GF(2^3)$ α^2 is primitive as can be easily shown. For convenience arbitrary field elements are represented by β. If we let $\beta = \alpha^2$, then constructing successive powers of β gives

$$\beta = \alpha^2$$
$$\beta^2 = (\alpha^2)^2 = \alpha^4$$
$$\beta^3 = (\alpha^2)^3 = \alpha^6$$
$$\beta^4 = (\alpha^2)^4 = \alpha^8 = \alpha \quad \text{(recall that } \alpha^7 = 1 \text{ in } GF(2^3)\text{)}$$
$$\beta^5 = (\alpha^2)^5 = \alpha^{10} = \alpha^3$$
$$\beta^6 = (\alpha^2)^6 = \alpha^{12} = \alpha^5.$$

So far this has generated six different elements, taking the next power gives

$$\beta^7 = (\alpha^2)^7 = \alpha^{14} = 1$$

and so further multiples of β will produce existing elements

$$\beta^8 = \beta = \alpha^2$$
$$\beta^9 = \beta^2 = \alpha^4$$

and so forth. Hence α^2 can also generate the nonzero elements of $GF(2^3)$ and is therefore a primitive field element of $GF(2^3)$. In fact all the elements (other than 0 and 1) of $GF(2^3)$ are primitive and therefore capable of generating the other nonzero elements.

Example 6.8
Show that α^5 is a primitive element of $GF(2^3)$.

Let $\beta = \alpha^5$ then

$$\beta^2 = \alpha^{10} = \alpha^3$$
$$\beta^3 = \alpha^{15} = \alpha$$
$$\beta^4 = \alpha^{20} = \alpha^6$$
$$\beta^5 = \alpha^{25} = \alpha^4$$
$$\beta^6 = \alpha^{30} = \alpha^2$$
$$\beta^7 = \alpha^{35} = 1.$$

Hence all 7 nonzero elements have been generated and α^5 is therefore primitive in $GF(2^3)$. □

We can likewise show that α^2 is primitive in $GF(2^4)$. However, consider next α^3 in $GF(2^4)$ and let $\beta = \alpha^3$, then

$$\beta = \alpha^3$$
$$\beta^2 = \alpha^6$$
$$\beta^3 = \alpha^9$$
$$\beta^4 = \alpha^{12}.$$

So far this has generated different elements, but the next term gives

$$\beta^5 = \alpha^{15} = 1 \text{(recall that } \alpha^{15} = 1 \text{ in } GF(2^4))$$

and therefore none of the remaining nonzero elements of $GF(2^4)$ can be generated. Continuing to take further powers of β will only generate $1, \alpha^3, \alpha^6, \alpha^9$, and α^{12}. For example

$$\beta^6 = \alpha^{18} = \alpha^3$$
$$\beta^7 = \alpha^{21} = \alpha^6$$
$$\beta^8 = \alpha^{24} = \alpha^9$$
$$\beta^9 = \alpha^{27} = \alpha^{12}$$
$$\beta^{10} = \alpha^{30} = 1.$$

Therefore within $GF(2^4)$ the field element α^3 is not primitive. There are other elements within $GF(2^4)$ that are not primitive, for example α^5 can only generate the elements 1 and α^{10}.

Whether or not a field element is primitive can be established by determining the *order* of the element, which for an element β is defined as the smallest positive integer n such that $\beta^n = 1$. This should not be confused with the order of a field, which is the number of elements within the field. In $GF(2^3)$ all the field elements have the same order 7. For example consider α^3 this has an order of 7 because $(\alpha^3)^7 = \alpha^{21} = 1$ and no other smaller power of α^3 gives 1. In $GF(2^4)$ however, not all elements have the same order. For example the order of α^5 is 3, whereas α^2 has an order of 15. The order of an element in $GF(2^m)$ divides $2^m - 1$ and furthermore determines whether or not the element is primitive. In a field $GF(2^m)$ a nonzero field element β is primitive if the order of β is $2^m - 1$. Within $GF(2^3)$ primitive field elements therefore have an order of 7 and primitive elements within $GF(2^4)$ have an order of 15.

Example 6.9
Given that α^{12} and α^7 are field elements of $GF(2^4)$ determine their order, whether or not they are primitive and the field elements generated if they are not primitive.

The smallest power of α^{12} to give unity is 5, as this gives $(\alpha^{12})^5 = \alpha^{60} = 1$. Hence α^{12} is not primitive. The elements generated by α^{12} are $(\alpha^{12})^2 = \alpha^{24} = \alpha^9$, $(\alpha^{12})^3 = \alpha^{36} = \alpha^6$, and $(\alpha^{12})^4 = \alpha^{48} = \alpha^3$. The next power of α^{12} gives $\alpha^{60} = 1$ and therefore α^{12} only generates $1, \alpha^3, \alpha^6$, and α^9.

The field element α^7 has order 15 as $(\alpha^7)^{15} = \alpha^{105} = 1$ and no smaller power of α^7 gives unity, it is therefore primitive and generates all the field elements of $GF(2^4)$. □

6.5 Irreducible and primitive polynomials

The polynomials $x^3 + x + 1$, $x^4 + x + 1$, and $x^5 + x^2 + 1$ used to generate $GF(2^3)$, $GF(2^4)$, and $GF(2^5)$ respectively cannot be factorized. Each polynomial is divisible only by itself and 1, such polynomials are referred to as *irreducible polynomials*. An irreducible polynomial having a primitive field element as a root is called a *primitive polynomial*. We have seen that $x^3 + x + 1$, $x^4 + x + 1$ and $x^5 + x^2 + 1$ have the primitive element α as a root and therefore the polynomials used to generate $GF(2^3)$, $GF(2^4)$, and $GF(2^5)$ are primitive.

For any positive integer m there is at least one irreducible polynomial of degree m. It can be shown that an irreducible polynomial of degree m divides $x^r + 1$ where $r = 2^m - 1$, and this can be used to establish whether or not a polynomial is irreducible. Take for example $x^3 + x + 1$, this should divide $x^7 + 1$ for it to be irreducible. Dividing $x^7 + 1$ by $x^3 + x + 1$ gives the quotient $x^4 + x^2 + x + 1$ and zero remainder, and therefore $x^3 + x + 1$ is irreducible. The reader can likewise show that $x^4 + x + 1$ and $x^5 + x^2 + 1$ are irreducible.

It is not always so easy, however, to establish whether or not an irreducible polynomial is primitive. It can be shown that an irreducible polynomial of degree m is primitive if it divides $x^r + 1$ for no r less than $2^m - 1$. Hence the polynomial must divide $x^r + 1$ but not $x^{r-1} + 1$, $x^{r-2} + 1$ and so forth. Consider again $x^3 + x + 1$, we have seen that it divides $x^7 + 1$ which shows that it is irreducible. To further show

that it is primitive we need to show that it does not divide x^6+1, x^5+1 or x^4+1 (there is no need to consider division into x^3+1, x^2+1 or $x+1$ because a polynomial of degree m cannot divide a polynomial of degree $\leq m$). Taking x^6+1 and dividing by x^3+x+1 gives

$$\begin{array}{r} x^3+x+1 \\ x^3+x+1\overline{)\,x^6+1} \\ \underline{x^6+x^4+x^3} \\ -\;x^4+x^3+1 \\ \underline{x^4+x^2+x} \\ -\;x^3+x^2+x+1 \\ \underline{x^3+x+1} \\ -\;x^2 \end{array}$$

resulting in a nonzero remainder. Likewise x^3+x+1 does not divide x^5+1 or x^4+1 and therefore x^3+x+1 is primitive.

The condition given for determining whether an irreducible polynomial is primitive is of limited use. However, a special case arises if 2^m-1 is prime, for an irreducible polynomial of degree m is primitive if 2^m-1 is prime. Care needs to be exercised here, because this special case means that if 2^m-1 is prime then the irreducible polynomial is primitive, but if 2^m-1 is not prime the polynomial may still be primitive. To illustrate this let's consider the polynomials used to generate $GF(2^3)$ and $GF(2^4)$. The field $GF(2^3)$ was generated using x^3+x+1 which has degree $m=3$, therefore $2^m-1=7$ is prime and so x^3+x+1 is primitive (as we have already seen). The polynomial x^4+x+1 used to generate $GF(2^4)$ has $m=4$ and $2^m-1=15$ which is not prime. However, the roots of the polynomial are α, α^2, α^4, and α^8 which are primitive elements of $GF(2^4)$ and therefore x^4+x+1 is primitive (recall that an irreducible polynomial with a primitive root is primitive).

Primitive polynomials are a special type of irreducible polynomials. With regard to generating a finite field it is the irreducible characteristic of a polynomial that is of importance. In order to generate a finite field it is not necessary for a polynomial to be primitive, it must however be irreducible. Primitive polynomials are preferred because it is easier to generate a field from a primitive polynomial than from one that is not primitive. A primitive polynomial has primitive roots and the field can be generated by taking successive powers of any primitive root. If an irreducible polynomial is not primitive then its roots are not primitive and each root generates only a limited number of field elements. To determine the remaining field elements a primitive element must first be found (recall that every finite field has at least one primitive element). To illustrate this consider the polynomial

$$p(x) = x^4 + x^3 + x^2 + x + 1. \tag{6.13}$$

The degree of $p(x)$ is $m=4$ and so for $p(x)$ to be irreducible it must divide $x^{15}+1$, which indeed it does. Hence $p(x)$ is irreducible and can be used to generate a field with 15 nonzero elements. Whether or not the polynomial is primitive cannot be determined from its degree because $2^m-1=15$ is not a prime number. To establish whether or not $p(x)$ is primitive we need to determine if it divides into $x^{14}+1$, $x^{13}+1,\ldots,$ or x^5+1. If $p(x)$ divides into any of these polynomials then it is not

primitive. It can be shown that $p(x)$ divides into x^5+1 and $p(x)$ is therefore not primitive. To construct the field generated by eqn 6.13 we let α be a root of $p(x)$ then $\alpha^4+\alpha^3+\alpha^2+\alpha+1=0$ and so

$$\alpha^4 = \alpha^3 + \alpha^2 + \alpha + 1.$$

Proceeding as usual to take successive powers of α gives:

$$\begin{aligned} &\alpha \\ &\alpha^2 \\ &\alpha^3 \\ &\alpha^4 = \alpha^3+\alpha^2+\alpha+1 \\ &\alpha^5 = \alpha\alpha^4 = \alpha(\alpha^3+\alpha^2+\alpha+1) = \alpha^4+\alpha^3+\alpha^2+\alpha = 1. \end{aligned} \tag{6.14}$$

Hence α has order 5 and is therefore not primitive and fails to generate the 15 nonzero elements of the field. To proceed further we need to find a primitive element. None of the elements α^2, α^3, or α^4 are primitive, taking any one of these will only generate the existing elements. Instead we need to consider some other element. If we let $\beta=\alpha+1$ we find that β is primitive with $\beta, \beta^2, \beta^3, \ldots, \beta^{15}$ giving the 15 nonzero field elements shown in Table 6.7. We have now constructed two finite fields containing 16 elements, the field generated by $x^4+x^3+x^2+x+1$ (Table 6.7) and that generated by x^4+x+1 (Table 6.4). However the two fields are just different representations of the same field $GF(2^4)$, because two finite fields with the same number of elements differ only in the way that the elements are labelled or ordered. Two finite fields with the same number of elements are said to be *isomorphic*, whilst they may have different representations their mathematical structure is the same.

Table 6.7
$GF(2^4)$ generated by $p(x)=x^4+x^3+x^2+x+1$

0
$\beta=\alpha+1$
$\beta^2=\alpha^2+1$
$\beta^3=\alpha^3+\alpha^2+\alpha+1$
$\beta^4=\alpha^3+\alpha^2+\alpha$
$\beta^5=\alpha^3+\alpha^2+1$
$\beta^6=\alpha^3$
$\beta^7=\alpha^2+\alpha+1$
$\beta^8=\alpha^3+1$
$\beta^9=\alpha^2$
$\beta^{10}=\alpha^3+\alpha^2$
$\beta^{11}=\alpha^3+\alpha+1$
$\beta^{12}=\alpha$
$\beta^{13}=\alpha^2+\alpha$
$\beta^{14}=\alpha^3+\alpha$
$\beta^{15}=1$

6.6 Minimal polynomials

We have seen that irreducible polynomials and primitive polynomials are used to construct finite fields. Here we consider minimal polynomials, which are used in the construction of binary codes (see Chapter 7).

Complex roots of equations with real coefficients always occur in pairs of complex conjugates. If $p + jq$ is a root of an equation with real coefficients then its complex conjugate $p - jq$ is also a root. The roots of a polynomial with binary coefficients likewise occur in conjugates, not necessarily in pairs but in groups or *sets of conjugates*. Given that β is a field element of $GF(2^m)$ then the conjugates of β are

$$\beta, \beta^2, \beta^4, \beta^8, \ldots, \beta^{2^{r-1}}$$

where r is the smallest integer such that $\beta^{2^r} = \beta$. For example consider the conjugates of α^5 in $GF(2^4)$

$$(\alpha^5)^2 = \alpha^{10}$$
$$(\alpha^5)^4 = \alpha^{20} = \alpha^5$$

therefore in $GF(2^4)$ α^5 has only one conjugate, α^{10}. The conjugates of α^7 in $GF(2^4)$ are

$$(\alpha^7)^2 = \alpha^{14}$$
$$(\alpha^7)^4 = \alpha^{28} = \alpha^{13}$$
$$(\alpha^7)^8 = \alpha^{56} = \alpha^{11}$$
$$(\alpha^7)^{16} = \alpha^{112} = \alpha^7$$

and therefore α^7 has the conjugates α^{11}, α^{13}, and α^{14}.

Example 6.10
Determine the conjugates of α^3 in $GF(2^3)$ and in $GF(2^4)$.

In $GF(2^4)$ we have:

$$(\alpha^3)^2 = \alpha^6$$
$$(\alpha^3)^4 = \alpha^{12}$$
$$(\alpha^3)^8 = \alpha^{24} = \alpha^9$$
$$(\alpha^3)^{16} = \alpha^{48} = \alpha^3$$

and therefore the conjugates of α^3 are α^6, α^9 and α^{12}. Whereas in $GF(2^3)$ the conjugates of α^3 are:

$$(\alpha^3)^2 = \alpha^6$$
$$(\alpha^3)^4 = \alpha^{12} = \alpha^5$$
$$(\alpha^3)^8 = \alpha^{24} = \alpha^3.$$

Note that the set of conjugates of α^3 in $GF(2^3)$ is different from that in $GF(2^4)$. □

Table 6.8
Conjugate elements in $GF(2^4)$ and in $GF(2^3)$

(a) $GF(2^4)$ Conjugates	Order	(b) $GF(2^3)$ Conjugates	Order
1	1	1	1
$\alpha, \alpha^2, \alpha^4, \alpha^8$	15	$\alpha, \alpha^2, \alpha^4$	7
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	5	$\alpha^3, \alpha^5, \alpha^6$	7
α^5, α^{10}	3		
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	15		

Table 6.8 shows the sets of conjugate elements in $GF(2^3)$ and $GF(2^4)$ along with the order of elements in the same conjugate set. Note that conjugate elements have the same order and therefore if an element β of $GF(2^m)$ is primitive then its conjugates are also primitive. Recall that the order of an element divides $2^m - 1$ and therefore if $2^m - 1$ is prime the field elements will have order $2^m - 1$ and be primitive. If $2^m - 1$ is not prime, then some elements will be nonprimitive with order less than $2^m - 1$ but there will be at least 1 primitive element. In $GF(2^3)$ $2^3 - 1 = 7$ is prime and therefore the field elements have order 7 and are primitive. $GF(2^4)$ has $2^4 - 1 = 15$ which is not prime and therefore the field has some nonprimitive elements with order less than 15.

One of the properties of conjugates is that they provide a mechanism for going from an extension field to its base field. Consider the pair of complex conjugates $z = p + jq$ and $z^* = p - jq$, their product gives the real number

$$zz^* = p^2 + q^2.$$

Taking the product of the two factors $(x - z)$ and $(x - z^*)$ likewise gives a real expression

$$(x - z)(x - z^*) = x^2 - 2px + p^2 + q^2.$$

In finite fields sets of conjugate elements perform the same function. Consider α^7, belonging to $GF(2^4)$, and its conjugates α^{11}, α^{13}, and α^{14}. Let

$$m(x) = (x + \alpha^7)(x + \alpha^{11})(x + \alpha^{13})(x + \alpha^{14})$$

then

$$\begin{aligned} m(x) &= (x^2 + x(\alpha^7 + \alpha^{11}) + \alpha^{18})(x^2 + x(\alpha^{13} + \alpha^{14}) + \alpha^{27}) \\ &= (x^2 + \alpha^8 x + \alpha^3)(x^2 + \alpha^2 x + \alpha^{12}) \\ &= x^4 + x^3(\alpha^2 + \alpha^8) + x^2(\alpha^{12} + \alpha^{10} + \alpha^3) + x(\alpha^{20} + \alpha^5) + \alpha^{15} \\ &= x^4 + x^3 + 1 \end{aligned}$$

which is a polynomial in the base field $GF(2)$. The polynomial $m(x)$ is referred to as the *minimal polynomial* of α^7, α^{11}, α^{13}, and α^{14}. It is the binary polynomial of smallest degree that has α^7, α^{11}, α^{13}, and α^{14} as roots. Let $m_i(x)$ denote the minimal polynomial of α^i, then $m_i(x)$ is defined to be the smallest degree polynomial in $GF(2)$

that has α^i as a root, and so

$$m_i(\alpha^i) = 0. \tag{6.15}$$

The minimal polynomial $m_i(x)$ is also the minimal polynomial of the conjugates of α and therefore

$$m_7(x) = m_{11}(x) = m_{13}(x) = m_{14}(x) = x^4 + x^3 + 1$$

where $m_7(x)$, $m_{11}(x)$, $m_{13}(x)$, and $m_{14}(x)$ are the minimal polynomials of α^7, α^{11}, α^{13}, and α^{14} respectively.

To determine the minimal polynomial $m(x)$ of an element β, a factor $(x + \beta^*)$ is required for each conjugate β^* of β. This ensures that the conjugate β^* is a root of $m(x)$. The minimal polynomial is then given by the product of all such factors, so that

$$m(x) = (x + \beta)(x + \beta^2)(x + \beta^4)\ldots(x + \beta^{2^{r-1}}) \tag{6.16}$$

where r is the smallest integer such that $\beta^{2^r} = \beta$. In $GF(2^4)$ the conjugates of α are α^2, α^4, α^8 and the minimal polynomial of α is therefore

$$\begin{aligned} m_1(x) &= (x + \alpha)(x + \alpha^2)(x + \alpha^4)(x + \alpha^8) \\ &= (x^2 + \alpha^5 x + \alpha^3)(x^2 + \alpha^5 x + \alpha^{12}) \\ &= x^4 + x + 1. \end{aligned}$$

The minimal polynomials of α^2, α^4, and α^8 are all equal to $m_1(x)$

$$m_2(x) = m_4(x) = m_8(x) = m_1(x) = x^4 + x + 1.$$

Table 6.9 gives the minimal polynomials of field elements in $GF(2^3)$ and $GF(2^4)$.

Table 6.9
Minimal polynomials in $GF(2^3)$ and $GF(2^4)$

Field elements	*Minimal polynomials*
(a) $GF(2^3)$	
0	x
1	$x+1$
$\alpha, \alpha^2, \alpha^4$	x^3+x+1
$\alpha^3, \alpha^5, \alpha^6$	x^3+x^2+1
(b) $GF(2^4)$	
0	x
1	$x+1$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	x^4+x+1
α^5, α^{10}	x^2+x+1
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$x^4+x^3+x^2+x+1$
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	x^4+x^3+1

Example 6.11
Find the minimal polynomial of α^5 in $GF(2^3)$.

The conjugates of α^5 are $(\alpha^5)^2 = \alpha^3$ and $(\alpha^5)^4 = \alpha^6$. The minimal polynomial of α^5 is therefore

$$\begin{aligned} m_5(x) &= (x + \alpha^5)(x + \alpha^3)(x + \alpha^6) \\ &= (x^2 + \alpha^2 x + \alpha)(x + \alpha^6) \\ &= x^3 + x^2(\alpha^6 + \alpha^2) + x(\alpha^8 + \alpha) + \alpha^7 \\ &= x^3 + x^2 + 1. \end{aligned}$$

This is also the minimal polynomial of α^3 and α^6. □

Consider the field element α^i and its conjugate α^{2i}, conjugate elements have the same minimal polynomial and therefore $m_{2i} = m_i$. Furthermore given that any even integer can be expressed as $2i$ where i is a smaller odd integer, we see therefore that the minimal polynomial of an even power of a field element is always equal to the minimal polynomial of some odd lower power of the field element.

6.7 Solution of equations in $GF(2^4)$ and $GF(2^3)$

We have seen that addition and multiplication of field elements can be carried out in finite fields. The finite fields though are not restricted to just addition and multiplication of field elements, for much of the mathematics that can be performed in the real and complex fields can also be performed in finite fields. Indeed it is often easier to carry out a mathematical operation in a finite field than in the real or complex fields due to the finite number of elements. Here we consider various characteristics of Galois fields, with particular reference to the solution of equations within the fields.

Consider first the linear equation

$$\alpha^3 x + \alpha^{11} = 0$$

defined in $GF(2^4)$. This differs from the equations previously considered in that its coefficients belong to an extension field. In the previous sections the coefficients of equations were real or binary, only the roots of equations were in extension fields. Nevertheless there is no reason why we cannot construct an equation whose coefficients belong to an extension field. We can easily solve the above equation, taking α^{11} over to the right-hand side gives $\alpha^3 x = \alpha^{11}$ and dividing through by α^3 gives $x = \alpha^{11}/\alpha^3 = \alpha^8$.

Every element in the field $GF(2^m)$ has a square root within $GF(2^m)$. Consider α^4 in $GF(2^4)$, its square root is $\sqrt{\alpha^4} = \alpha^2$. Likewise in $GF(2^3)$ the square root of α^4 is α^2. However the square root of α^5 in $GF(2^4)$ is not so obvious. To deal with this we multiply α^5 by α^{15} and then take the square root, so giving

$$\sqrt{\alpha^5} = \sqrt{(\alpha^5 \alpha^{15})} = \sqrt{\alpha^{20}} = \alpha^{10}.$$

Note, however, that in $GF(2^3)$ the square root of α^5 is

$$\sqrt{\alpha^5} = \sqrt{(\alpha^5 \alpha^7)} = \sqrt{\alpha^{12}} = \alpha^6.$$

Hence the square root of α^5 in $GF(2^3)$ differs from its square root in $GF(2^4)$.

Next consider the roots of the quadratic equation

$$x^2 + \alpha^{12}x + \alpha^9 = 0$$

defined over $GF(2^4)$. Here again we have an equation whose coefficients belong to an extension field. We can factorize the above equation by using the standard approach of establishing two terms whose sum and product give the required coefficients. If β_1 and β_2 are the required roots then

$$(x + \beta_1)(x + \beta_2) = 0$$

and expanding this gives

$$x^2 + x(\beta_1 + \beta_2) + \beta_1\beta_2 = 0$$

and therefore we need to find the field elements β_1 and β_2 that satisfy

$$\beta_1 + \beta_2 = \alpha^{12}$$
$$\beta_1\beta_2 = \alpha^9.$$

Referring to Table 6.5 we see that the field elements α^2 and α^7 meet this requirement, since

$$\alpha^2\alpha^7 = \alpha^9$$
$$\alpha^2 + \alpha^7 = \alpha^{12}$$

and therefore

$$(x + \alpha^2)(x + \alpha^7) = x^2 + \alpha^{12}x + \alpha^9 = 0$$

so giving α^2 and α^7 as the roots. The solution of equations in a finite field can be achieved by a trial-and-error method in which field elements are systematically tested to see if they are roots. Such an approach of searching for roots is referred to as a *Chien search*. For example consider the roots of $p(x) = x^3 + \alpha^9x^2 + \alpha^6x + \alpha^2$ over $GF(2^4)$. Starting with $x = 0$ gives

$$p(0) = \alpha^2$$
$$p(1) = \alpha^4$$
$$p(\alpha) = \alpha^{14}$$
$$p(\alpha^2) = 0$$

and so $x = \alpha^2$ is one of the roots. Continuing with the search shows that the other roots are α^7 and α^8. Note that within an extension field polynomials can exist that do not have roots within the field but lie within some other field. Consider for example

$$p(x) = x^2 + \alpha^2x + \alpha^{10}$$

in $GF(2^4)$. A search fails to find any roots and therefore $p(x)$ is irreducible over $GF(2^4)$.

A useful property of the field $GF(2^m)$ is that the square of a series of terms added together is equal to sum of the individual terms squared. Consider $x_1 + x_2$ squared

$$(x_1 + x_2)^2 = x_1^2 + x_1x_2 + x_2x_1 + x_2^2$$

and because $x_1x_2 + x_2x_1 = 2x_1x_2 = 0$ we see that

$$(x_1 + x_2)^2 = x_1^2 + x_2^2$$

Squaring this again gives

$$\{(x_1 + x_2)^2\}^2 = \{(x_1^2 + x_2^2)\}^2 = x_1^4 + x_2^4$$

and so

$$(x_1 + x_2)^4 = x_1^4 + x_2^4$$

This can be extended to all powers 2^i, where i is a positive integer, of $(x_1 + x_2)$ so giving

$$(x_1 + x_2)^{2^i} = x_1^{2^i} + x_2^{2^i}. \tag{6.17}$$

For example in $GF(2^4)$

$$\begin{aligned}(x + \alpha^7)^8 &= x^8 + (\alpha^7)^8\\ &= x^8 + \alpha^{56}\\ &= x^8 + \alpha^{11}.\end{aligned}$$

Care must be taken not to incorrectly apply eqn 6.17, for instance $(x_1 + x_2)^6 \neq (x_1^6 + x_2^6)$. Equation 6.17 can, though, still be used to expand such an expression

$$\begin{aligned}(x_1 + x_2)^6 &= (x_1 + x_2)^4(x_1 + x_2)^2\\ &= (x_1^4 + x_2^4)(x_1^2 + x_2^2)\\ &= x_1^6 + x_2^2x_1^4 + x_2^4x_1^2 + x_2^6.\end{aligned}$$

Example 6.12

Expand (a) $(x + \alpha^4)^2$ in $GF(2^3)$ and (b) $(x + \alpha^3)^5(x + \alpha^{10})$ in $GF(2^4)$.

(a) In $GF(2^3)$ we have $(x + \alpha^4)^2 = x^2 + \alpha^8 = x^2 + \alpha$.

(b) In $GF(2^4)$

$$\begin{aligned}(x + \alpha^3)^5(x + \alpha^{10}) &= (x + \alpha^3)^4(x + \alpha^3)(x + \alpha^{10})\\ &= (x^4 + \alpha^{12})(x^2 + \alpha^{12}x + \alpha^{13})\\ &= x^6 + \alpha^{12}x^5 + \alpha^{13}x^4 + \alpha^{12}x^2 + \alpha^9x + \alpha^{10}\end{aligned}$$ □

Equation 6.17 can be applied to a series of r terms, given $x_1 + x_2 + \cdots + x_r$ in $GF(2^m)$ then

$$(x_1 + x_2 + \cdots + x_r)^{2^i} = x_1^{2^i} + x_2^{2^i} + \cdots + x_r^{2^i} \tag{6.18}$$

where i is a positive integer.

Matrices and determinants of field elements can be constructed and are subject to the same algebraic rules as when constructed with real or complex numbers, obviously though using the additive and multiplicative rules of the finite field within which the field elements exist. Over a field $GF(2^m)$ the matrix

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rn} \end{bmatrix}$$

can be defined where r and n are positive integers, and where a_{ij} are field elements with $i = 1, 2, \ldots, r$ and $j = 1, 2, \ldots, n$. Consider the 3 by 3 matrix defined in $GF(2^4)$

$$\boldsymbol{A}_1 = \begin{bmatrix} \alpha^2 & \alpha & \alpha^{13} \\ 0 & \alpha^{10} & \alpha \\ \alpha^7 & \alpha^3 & 1 \end{bmatrix}.$$

The determinant of $\boldsymbol{A}_1$ is

$$\begin{aligned} \det \boldsymbol{A}_1 &= \alpha^2 \begin{vmatrix} \alpha^{10} & \alpha \\ \alpha^3 & 1 \end{vmatrix} + \alpha \begin{vmatrix} 0 & \alpha \\ \alpha^7 & 1 \end{vmatrix} + \alpha^{13} \begin{vmatrix} 0 & \alpha^{10} \\ \alpha^7 & \alpha^3 \end{vmatrix} \\ &= \alpha^2(\alpha^{10} + \alpha^4) + \alpha(0 + \alpha^8) + \alpha^{13}(0 + \alpha^{17}) \\ &= \alpha^2\alpha^2 + \alpha\alpha^8 + \alpha^{13}\alpha^2 = \alpha^4 + \alpha^9 + 1 = \alpha^3. \end{aligned}$$

Note that $\det \boldsymbol{A}_1$ is a field element of $GF(2^4)$. As $\det \boldsymbol{A}_1$ is nonzero we can determine the inverse of $\boldsymbol{A}_1$. The inverse $\boldsymbol{A}^{-1}$ of a square matrix $\boldsymbol{A}$ is given by $\text{adj}\,\boldsymbol{A}/\det \boldsymbol{A}$ where $\text{adj}\,\boldsymbol{A}$ is the adjoint of $\boldsymbol{A}$ and $\det \boldsymbol{A} \neq 0$. The adjoint of a matrix is the transpose of the matrix formed from the cofactors of $\boldsymbol{A}$ and so

$$\text{adj}\,\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{r1} \\ A_{12} & A_{22} & \cdots & A_{r2} \\ \vdots & & & \vdots \\ A_{1r} & A_{2r} & \cdots & A_{rr} \end{bmatrix}$$

where the cofactor A_{ij} is the determinant constructed by excluding the row and column that a_{ij} lies in. A plus or minus sign is normally attached to the cofactor, however as the base field is $GF(2)$ there is no need for this. To find the inverse of $\boldsymbol{A}_1$ we first determine the cofactors

$$A_{11} = \begin{vmatrix} \alpha^{10} & \alpha \\ \alpha^3 & 1 \end{vmatrix} = \alpha^{10}1 + \alpha\alpha^3 = \alpha^{10} + \alpha^4 = \alpha^2$$

$$A_{21} = \begin{vmatrix} \alpha & \alpha^{13} \\ \alpha^3 & 1 \end{vmatrix} = \alpha 1 + \alpha^{13}\alpha^3 = \alpha + \alpha = 0$$

and continuing to evaluate the other 7 cofactors we get

$$\text{adj}\,\boldsymbol{A}_1 = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix} = \begin{bmatrix} \alpha^2 & 0 & 1 \\ \alpha^8 & \alpha & \alpha^3 \\ \alpha^2 & \alpha^4 & \alpha^{12} \end{bmatrix}$$

The inverse of $\boldsymbol{A}_1$, is therefore

$$\begin{aligned} \boldsymbol{A}_1^{-1} &= \text{adj}\,\boldsymbol{A}_1/\det \boldsymbol{A}_1 \\ &= (1/\alpha^3)\begin{bmatrix} \alpha^2 & 0 & 1 \\ \alpha^8 & \alpha & \alpha^3 \\ \alpha^2 & \alpha^4 & \alpha^{12} \end{bmatrix} \\ &= \begin{bmatrix} \alpha^2/\alpha^3 & 0 & 1/\alpha^3 \\ \alpha^8/\alpha^3 & \alpha/\alpha^3 & \alpha^3/\alpha^3 \\ \alpha^2/\alpha^3 & \alpha^4/\alpha^3 & \alpha^{12}/\alpha^3 \end{bmatrix} \end{aligned}$$

which gives

$$\boldsymbol{A}_1^{-1} = \begin{bmatrix} \alpha^{14} & 0 & \alpha^{12} \\ \alpha^5 & \alpha^{13} & 1 \\ \alpha^{14} & \alpha & \alpha^9 \end{bmatrix}$$

The reader can check that $\boldsymbol{A}_1\boldsymbol{A}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \boldsymbol{I}$

Example 6.13
Determine the inverse of the matrix

$$\boldsymbol{A} = \begin{bmatrix} \alpha^3 & \alpha^5 \\ 1 & \alpha \end{bmatrix}$$

in $GF(2^3)$ and $GF(2^4)$.

In $GF(2^3)$ the determinant of $\boldsymbol{A}$ is

$$\det \boldsymbol{A} = \begin{vmatrix} \alpha^3 & \alpha^5 \\ 1 & \alpha \end{vmatrix} = \alpha^3\alpha + \alpha^5 1 = \alpha^4 + \alpha^5 = 1$$

As $\det \boldsymbol{A} \neq 0$ the inverse therefore exists. The cofactors of $\boldsymbol{A}$ are $A_{11} = \alpha$, $A_{12} = 1$, $A_{21} = \alpha^5$ and $A_{22} = \alpha^3$ and so the adjoint of $\boldsymbol{A}$ is

$$\text{adj}\,\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{bmatrix} = \begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^3 \end{bmatrix}$$

The inverse of $\boldsymbol{A}$ is therefore

$$\boldsymbol{A}^{-1} = (\text{adj}\,\boldsymbol{A})/\det \boldsymbol{A} = \begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^3 \end{bmatrix}$$

We can check that

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \begin{bmatrix} \alpha^3 & \alpha^5 \\ 1 & \alpha \end{bmatrix}\begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^3 \end{bmatrix} = \begin{bmatrix} \alpha^4+\alpha^5 & \alpha^8+\alpha^8 \\ \alpha+\alpha & \alpha^5+\alpha^4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

as required.

In $GF(2^4)$ we get

$$\det \boldsymbol{A} = \alpha^8$$

$$\text{adj } \boldsymbol{A} = \begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^3 \end{bmatrix}$$

which gives

$$\boldsymbol{A}^{-1} = \begin{bmatrix} \alpha^8 & \alpha^{12} \\ \alpha^7 & \alpha^{10} \end{bmatrix}. \qquad \square$$

Linear equations can be defined over finite fields and solved using standard methods. For example, take

$$\alpha x + \alpha^5 y = \alpha^3$$
$$x + \alpha^7 y = \alpha^{11}$$

defined over $GF(2^4)$. These can be solved in the normal manner of first multiplying one of the equations by a suitable number and then subtracting the equations, thus eliminating one variable. Here multiplying $x + \alpha^7 y = \alpha^{11}$ by α and then adding it to $\alpha x + \alpha^5 y = \alpha^3$ eliminates the x variable so leaving

$$(\alpha^5 + \alpha^8)y = \alpha^3 + \alpha^{12}$$

which is easily solved to give $y = \alpha^6$. Substituting this into either of the simultaneous equations gives $x = \alpha^4$.

Matrix inversion techniques can also be used to solve linear equations in finite fields, adopting this approach the above linear equations can be written as

$$\begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^7 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha^3 \\ \alpha^{11} \end{bmatrix}$$

with the solution given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^7 \end{bmatrix}^{-1}\begin{bmatrix} \alpha^3 \\ \alpha^{11} \end{bmatrix}.$$

Here we get

$$\text{adj}\begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^7 \end{bmatrix} = \begin{bmatrix} \alpha^7 & \alpha^5 \\ 1 & \alpha \end{bmatrix}$$

and

$$\begin{vmatrix} \alpha & \alpha^5 \\ 1 & \alpha^7 \end{vmatrix} = \alpha\alpha^7 + 1\alpha^5 = \alpha^4$$

so that the required inverse matrix is

$$\begin{bmatrix} \alpha & \alpha^5 \\ 1 & \alpha^7 \end{bmatrix}^{-1} = \frac{1}{\alpha^4}\begin{bmatrix} \alpha^7 & \alpha^5 \\ 1 & \alpha \end{bmatrix} = \begin{bmatrix} \alpha^7/\alpha^4 & \alpha^5/\alpha^4 \\ 1/\alpha^4 & \alpha/\alpha^4 \end{bmatrix} = \begin{bmatrix} \alpha^3 & \alpha \\ \alpha^{11} & \alpha^{12} \end{bmatrix}$$

Therefore the solution is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha^3 & \alpha \\ \alpha^{11} & \alpha^{12} \end{bmatrix}\begin{bmatrix} \alpha^3 \\ \alpha^{11} \end{bmatrix} = \begin{bmatrix} \alpha^3\alpha^3 + \alpha\alpha^{11} \\ \alpha^{11}\alpha^3 + \alpha^{12}\alpha^{11} \end{bmatrix} = \begin{bmatrix} \alpha^4 \\ \alpha^6 \end{bmatrix}$$

which gives $x = \alpha^4$, $y = \alpha^6$ as obtained previously. The next example considers a set of three linear equations.

Example 6.14

Using matrix inversion, determine the solution of the following set of linear equations over $GF(2^4)$:

$$\alpha^3 x_1 + \alpha x_2 + x_3 = \alpha^5$$
$$\alpha^2 x_1 + \alpha^6 x_2 + x_3 = \alpha^6$$
$$\alpha^{14} x_1 + \alpha^7 x_2 + \alpha^7 x_3 = 1.$$

Representing the equations in matrix form gives

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$$

where

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{c} = \begin{bmatrix} \alpha^5 \\ \alpha^6 \\ 1 \end{bmatrix}$$

are column vectors, and $\boldsymbol{A}$ is the matrix

$$\boldsymbol{A} = \begin{bmatrix} \alpha^3 & \alpha & 1 \\ \alpha^2 & \alpha^6 & 1 \\ \alpha^{14} & \alpha^7 & \alpha^7 \end{bmatrix}$$

The determinant of A is

$$\det A = \alpha^3\begin{vmatrix}\alpha^9 & 1\\ \alpha^7 & \alpha^7\end{vmatrix} + \alpha\begin{vmatrix}\alpha^2 & 1\\ \alpha^{14} & \alpha^7\end{vmatrix} + 1\begin{vmatrix}\alpha^2 & \alpha^9\\ \alpha^{14} & \alpha^7\end{vmatrix}$$
$$= \alpha^3(\alpha^{13} + \alpha^7) + \alpha(\alpha^9 + \alpha^{14}) + 1(\alpha^9 + \alpha^3)$$
$$= \alpha^5 + \alpha^3 + \alpha^8 = \alpha^{12}$$

The adjoint of A is

$$\text{adj } A = \begin{bmatrix}\alpha^5 & \alpha^{11} & \alpha^{13}\\ \alpha^4 & \alpha^{11} & \alpha^9\\ \alpha^9 & \alpha^5 & \alpha\end{bmatrix}$$

and using $A^{-1} = \text{adj } A/\det A$ gives

$$A^{-1} = \begin{bmatrix}\alpha^8 & \alpha^{14} & \alpha^{14}\\ \alpha^7 & \alpha^{14} & \alpha^8\\ \alpha^8 & \alpha^8 & \alpha^4\end{bmatrix}$$

Multiplying $Ax = c$ by A^{-1} gives $x = A^{-1}c$ and therefore

$$\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} = \begin{bmatrix}\alpha^8 & \alpha^{14} & \alpha^{14}\\ \alpha^7 & \alpha^{14} & \alpha^8\\ \alpha^8 & \alpha^8 & \alpha^4\end{bmatrix}\begin{bmatrix}\alpha^3\\ \alpha^8\\ 1\end{bmatrix}$$
$$= \begin{bmatrix}\alpha^{13} + \alpha^3 + \alpha^{14}\\ \alpha^{12} + \alpha^5 + \alpha^8\\ \alpha^{14} + \alpha^{14} + \alpha^4\end{bmatrix}$$
$$= \begin{bmatrix}\alpha\\ \alpha^4\\ \alpha^4\end{bmatrix}$$

which gives $x_1 = \alpha$, $x_2 = \alpha^4$, and $x_3 = \alpha^4$ □

Problems

6.1 Given that α is a field element of $GF(2^5)$ evaluate
(a) $(\alpha^2\alpha^{-5} + 1)(\alpha^4 + \alpha)$
(b) $\sqrt{(\alpha^4\alpha^5 + \sqrt{\alpha})}$.
Repeat when α is an element of $GF(2^4)$.

6.2 Determine whether the polynomials

$$p_1(x) = x^4 + x^3 + x + 1$$
$$p_2(x) = x^2 + x + 1$$
$$p_3(x) = x^3 + x^2 + 1$$

over $GF(2)$ are (a) irreducible and (b) primitive.

6.3 Given the polynomial $p(x) = x^2 + x + 1$ over $GF(2)$, construct the field $GF(2^2)$ and therefore find the roots of $p(x) = 0$.

6.4 Show that the field elements of $GF(2^3)$ are primitive (except for 0 and 1). Determine whether the elements α^7 and α^{12} in $GF(2^4)$ are primitive.

6.5 Given that β is a root of the irreducible polynomial $p(x) = x^3 + x^2 + 1$ over $GF(2)$, construct the field $GF(2^3)$ using $p(x)$. Note that the field $GF(2^3)$ constructed using $x^3 + x^2 + 1$ is the same as that constructed using $x^3 + x + 1$, they differ only in the way in which elements are labelled.

6.6 Determine the conjugate sets for the field elements of $GF(2^5)$ (see Table 6.6). Show that the minimal polynomials of α and α^3 in $GF(2^5)$ are

$$m_1(x) = x^5 + x^2 + 1$$
$$m_3(x) = x^5 + x^4 + x^3 + x^2 + 1$$

respectively.

6.7 Find the roots of $x^3 + \alpha^8x^2 + \alpha^{12}x + \alpha = 0$ defined over $GF(2^4)$.

6.8 Find the determinant of the matrix

$$A = \begin{bmatrix}1 & \alpha^4 & \alpha^3\\ \alpha^2 & 0 & \alpha\\ \alpha^4 & \alpha & \alpha^5\end{bmatrix}$$

over $GF(2^3)$ and $GF(2^4)$.

6.9 Determine the inverse of the matrix

$$A = \begin{bmatrix}\alpha^{12} & 0 & \alpha\\ 1 & \alpha^8 & \alpha^{14}\\ \alpha^2 & \alpha^{11} & \alpha^5\end{bmatrix}$$

over $GF(2^3)$ and $GF(2^4)$.

6.10 Solve the linear equations

$$x + \alpha^4y = \alpha^5$$
$$\alpha^5x + \alpha^2y = \alpha^3$$

defined over $GF(2^4)$. Show that the equations do not have a unique solution when defined over $GF(2^3)$.

6.11 Solve the linear equations

$$\alpha^{10}x + \alpha^4y + \alpha z = \alpha^9$$
$$\alpha^5x + \alpha^{12}y + z = \alpha^4$$
$$\alpha^2x + \alpha^4y + \alpha^7z = \alpha^7$$

defined over $GF(2^3)$. Repeat over $GF(2^4)$.

Bose–Chaudhuri–Hocquenghem codes

7

Having considered the properties of finite fields we are now in a good position to move on, from cyclic codes, to the next level of codes namely the *Bose–Chaudhuri–Hocquenghem* (BCH) codes. Cyclic codes were introduced from the point of view of their cyclic property. Later we saw that one of the properties of cyclic codes is that codewords have their generator polynomial as a factor and the roots of a code's generator polynomial are therefore roots of the codewords. Here we first reconsider cyclic codes in terms of roots in an extension field and then we see that by using a well-defined set of roots we can construct BCH codes. The BCH codes are a subset of cyclic codes, they are a powerful class of multiple-error correcting codes with well understood mathematical properties. Binary and nonbinary BCH codes exist, and in particular the *Reed–Solomon codes* are a popular nonbinary class of BCH codes which find many applications. BCH codes are generally considered to be the most important class of codes, a study of error-control codes is incomplete without considering BCH codes.

7.1 Cyclic codes revisited

We have seen that a codeword polynomial $c(x)$ of an (n, k) cyclic code can always be written as

$$c(x) = f(x)g(x) \tag{7.1}$$

where for a nonsystematic code $f(x)$ is the information polynomial $i(x)$, whilst for a systematic code $f(x)$ is the quotient $q(x)$ obtained by dividing $i(x)x^{n-k}$ by $g(x)$. From eqn 7.1, it is clear that any root of $g(x)$, is also a root of $c(x)$, for if we let β be a root of $g(x)$, then $g(\beta) = 0$ and

$$c(\beta) = f(\beta)g(\beta) = 0.$$

Hence in a cyclic code the roots of the generator polynomial $g(x)$ are also roots of the codeword polynomials. Consider the codeword polynomial

$$c(x) = x^5 + x^2 + x + 1$$

belonging to the (7, 4) cyclic code generated by $g(x) = x^3 + x + 1$. The 3 roots of $x^3 + x + 1$ are the field elements α, α^2, and α^4 belonging to $GF(2^3)$. Substituting $x = \alpha$ into $c(x)$ gives

$$c(\alpha) = \alpha^5 + \alpha^2 + \alpha + 1 = 0$$

where the calculations are carried out in $GF(2^3)$. Likewise if we let $x = \alpha^2$ then

$$c(\alpha^2) = \alpha^{10} + \alpha^4 + \alpha^2 + 1 = 0$$

and $x = \alpha^4$ gives

$$c(\alpha^4) = \alpha^{20} + \alpha^8 + \alpha^4 + 1 = 0$$

The reader may wish to verify that any of the other codeword polynomials of the (7, 4) code also have α, α^2, and α^4 as roots.

Next consider the (15, 11) cyclic code with the generator polynomial $g(x) = x^4 + x + 1$ and

$$c(x) = x^6 + x^4 + x^3 + x^2 + x + 1$$

as one of its codeword polynomials. The roots of $g(x)$ are α, α^2, α^4, and α^8 in $GF(2^4)$, and substituting these into $c(x)$ gives

$$c(\alpha) = \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1 = 0$$
$$c(\alpha^2) = \alpha^{12} + \alpha^8 + \alpha^6 + \alpha^4 + \alpha^2 + 1 = 0$$
$$c(\alpha^4) = \alpha^9 + \alpha + \alpha^{12} + \alpha^8 + \alpha^4 + 1 = 0$$
$$c(\alpha^8) = \alpha^3 + \alpha^2 + \alpha^9 + \alpha + \alpha^8 + 1 = 0$$

and so, yet again, the roots of the generator polynomial are roots of the codeword polynomial.

The polynomials $x^3 + x + 1$ and $x^4 + x + 1$ serve 2 functions, they are generator polynomials of cyclic codes and, because they are irreducible, they are used to construct finite fields. A generator polynomial need not be irreducible in which case it can not be used to construct a finite field. For example, the generator polynomial for the (15, 7) cyclic code

$$g(x) = x^8 + x^7 + x^6 + x^4 + 1$$

can be factorized as

$$g(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)$$

and is therefore reducible in $GF(2)$ and cannot be used to construct a finite field. The 8 roots of $g(x)$ are field elements in $GF(2^4)$ with minimal polynomials $x^4 + x + 1$ or $x^4 + x^3 + x^2 + x + 1$ but $g(x)$ cannot be used to construct $GF(2^4)$.

The generator polynomials $x^3 + x + 1$ and $x^4 + x + 1$ can be expressed as

$$x^3 + x + 1 = (x + \alpha)(x + \alpha^2)(x + \alpha^4)$$

and

$$x^4 + x + 1 = (x + \alpha)(x + \alpha^2)(x + \alpha^4)(x + \alpha^8)$$

and we think of each generator polynomial as being specified by a chosen set of field elements. So $x^3 + x + 1$ is specified by α, α^2 and α^4 in $GF(2^3)$, and $x^4 + x + 1$ by α, α^2, α^4, and α^8 in $GF(2^4)$. Note that α, α^2, and α^4 form a conjugate set in $GF(2^3)$ with minimal polynomial $x^3 + x + 1$, and that in $GF(2^4)$ α, α^2, α^4, and α^8 form a conjugate set with minimal polynomial $x^4 + x + 1$ (see Tables 6.8 and 6.9).

To construct a generator polynomial $g(x)$ from an arbitrary set of r field elements $\beta_1, \beta_2, \ldots, \beta_r$ we need to find the polynomial of least degree that has $\beta_1, \beta_2, \ldots, \beta_r$ as its roots. The polynomial

$$g(x) = (x+\beta_1)(x+\beta_2)\cdots(x+\beta_r) \tag{7.2}$$

satisfies this requirement, but may not be a binary polynomial. For example taking $\beta_1 = \alpha$ and $\beta_2 = \alpha^2$ in $GF(2^3)$ gives $g(x) = (x+\alpha)(x+\alpha^2) = x^2 + \alpha^4 x + \alpha^3$ which has coefficients in $GF(2^3)$. Replacing each factor $(x+\beta_i)$ by $m_i(x)$, the minimal polynomial of β_i, gives

$$g(x) = m_1(x)m_2(x)\cdots m_r(x) \tag{7.3}$$

which is now a binary polynomial with roots $\beta_1, \beta_2, \ldots, \beta_r$. If any of the elements $\beta_1, \beta_2, \ldots, \beta_r$ are conjugates of each other then eqn 7.3 will contain multiple factors of the conjugates' minimal polynomial. Taking the *Least Common Multiple* (LCM) of $g(x)$ will exclude all such common multiples and give

$$g(x) = \text{LCM}[m_1(x), m_2(x), \ldots, m_r(x)] \tag{7.4}$$

as the binary polynomial of least degree with roots $\beta_1, \beta_2, \ldots, \beta_r$. In $GF(2^m)$ the product of 2 or more minimal polynomials divides $x^{q-1}+1$, where $q = 2^m$, and therefore $g(x)$ as given by eqn 7.4 is a generator polynomial for a cyclic code.

Let's reconsider the (7, 4) code whose generator polynomial $x^3 + x + 1$ has roots α, α^2, and α^4 in $GF(2^3)$. According to eqn 7.4 the generator polynomial specified by α, α^2, and α^4 is

$$g(x) = \text{LCM}[m_1(x), m_2(x), m_4(x)]$$

where $m_1(x)$, $m_2(x)$, and $m_4(x)$ are the minimal polynomials of α, α^2 and α^4 respectively. However in $GF(2^3)$ $m_1(x) = m_2(x) = m_4(x) = x^3 + x + 1$ and therefore we get

$$g(x) = \text{LCM}[m_1(x), m_1(x), m_1(x)] = m_1(x) = x^3 + x + 1$$

as required.

7.2 Definition and construction of binary BCH codes

We have seen how to construct a generator polynomial, of a cyclic code, with an arbitrary set of field elements as its roots. The BCH codes are a subset of cyclic codes whose generator polynomials have roots carefully specified so as to give good error-correcting capability. A t-error-correcting cyclic code with generator polynomial $g(x)$ is a binary BCH code if and only if $g(x)$ is the least-degree polynomial over $GF(2)$ that has

$$\beta, \beta^2, \beta^3, \ldots, \beta^{2t}$$

as roots, where β is an element of $GF(2^m)$. It can be shown that with this selection of roots the resulting code is capable of correcting t errors. If the field element β is

primitive then the codes are known as *primitive BCH codes* and have a blocklength of $n = 2^m - 1$. The BCH codes considered here are primitive, unless stated otherwise, with

$$\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$$

as the $2t$ consecutive roots. Using eqn 7.4 we can see therefore that the generator polynomial $g(x)$ of a t-error-correcting binary BCH code is given by

$$g(x) = \text{LCM}[m_1(x), m_2(x), m_3(x), \ldots, m_{2t}(x)] \tag{7.5}$$

where $m_i(x)$ is the minimal polynomial of α^i and α is an element of $GF(2^m)$. As the minimal polynomial of an even power of a field element is always equal to the minimal polynomial of some odd and lower power of the element, the minimal polynomials with even i can be omitted from eqn 7.5 and therefore

$$g(x) = \text{LCM}[m_1(x), m_3(x), m_5(x), \ldots, m_{2t-1}(x)]. \tag{7.6}$$

The blocklength of a primitive BCH code constructed over $GF(2^m)$ is $n = 2^m - 1$. BCH codes are cyclic codes and the degree r of the generator polynomial of an (n, k) cyclic code is $n - k$. Hence the information length k of a BCH code is $k = 2^m - 1 - r$.

As an example let's construct a double-error-correcting BCH code over $GF(2^4)$. Here $t = 2$ and taking $\beta = \alpha$, where α is a primitive element of $GF(2^4)$, gives

$$g(x) = \text{LCM}[m_1(x), m_2(x), m_3(x), m_4(x)]$$

where $m_1(x)$, $m_2(x)$, $m_3(x)$ and $m_4(x)$ are the minimal polynomials of $\alpha, \alpha^2, \alpha^3$, and α^4 respectively. In $GF(2^4)$ the minimal polynomials

$$\begin{aligned} m_1(x) &= x^4 + x + 1 \\ m_2(x) &= m_1(x) \\ m_3(x) &= x^4 + x^3 + x^2 + x + 1 \\ m_4(x) &= m_1(x) \end{aligned}$$

and so $m_2(x)$ and $m_4(x)$ can be excluded from $g(x)$, therefore giving

$$\begin{aligned} g(x) &= m_1(x)m_3(x) \\ &= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) \\ &= x^8 + x^7 + x^6 + x^4 + 1. \end{aligned}$$

As α is a primitive element of $GF(2^4)$, the blocklength of the constructed code is $n = 2^m - 1 = 15$. The degree of $g(x)$ is $r = 8$ and the information length is $k = n - r = 7$. We have therefore constructed the double-error-correcting (15, 7) BCH code with $g(x) = x^8 + x^7 + x^6 + x^4 + 1$.

A single-error-correcting code with the same blocklength can be constructed over the same field. Let $t = 1$ then

$$g(x) = \text{LCM}[m_1(x), m_2(x)].$$

Over $GF(2^4)$ the minimal polynomial $m_2(x) = m_1(x) = x^4 + x + 1$ and therefore

$$g(x) = m_1(x) = x^4 + x + 1.$$

The blocklength is $n = 2^m - 1 = 15$ and the information length is $k = n - r = 11$ because the degree of $g(x)$ is $r = 4$. This is therefore the single-error-correcting (15, 11) BCH code with $g(x) = x^4 + x + 1$.

Example 7.1
Construct a triple-error-correcting BCH code with blocklength $n = 31$ over $GF(2^5)$.

Let $t = 3$ and α be a primitive element of $GF(2^5)$. The generator polynomial is

$$g(x) = \text{LCM}[m_1(x), m_2(x), m_3(x), m_4(x), m_5(x), m_6(x)].$$

In $GF(2^5)$

$$\begin{aligned} m_1(x) &= x^5 + x^2 + 1 \\ m_2(x) &= m_1(x) \\ m_3(x) &= x^5 + x^4 + x^3 + x^2 + 1 \\ m_4(x) &= m_2(x) \\ m_5(x) &= x^5 + x^4 + x^2 + x + 1 \\ m_6(x) &= m_3(x) \end{aligned}$$

and $g(x)$ therefore reduces to

$$\begin{aligned} g(x) &= m_1(x)m_3(x)m_5(x) \\ &= (x^5 + x^2 + 1)(x^5 + x^4 + x^3 + x^2 + 1)(x^5 + x^4 + x^2 + x + 1) \\ &= x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1. \end{aligned}$$

The blocklength and information length are $n = 2^5 - 1 = 31$ and $k = 31 - 15 = 16$ respectively. This is therefore the (31, 16) triple-error-correcting binary BCH code. □

A t-error-correcting BCH code has a guaranteed minimum distance of $d = 2t + 1$. However the minimum distance $d_{\min}$ of the code may be greater than d, therefore giving the code an error-control capability greater than that designed. As such the minimum distance $d = 2t + 1$ is known as the *designed distance* of the code, and for any BCH code $d_{\min} \geq d$.

BCH codes are cyclic and linear, and so once a code's generator polynomial $g(x)$ is constructed, encoding can be carried out in the usual manner using $g(x)$ or the generator matrix G constructed from $g(x)$. It is at the decoding stage that techniques specific to BCH codes are used.

7.3 Error syndromes in finite fields

At the decoding stage of an error-correcting code, decisions are made on the basis of error syndromes that depend on the presence of errors. Codewords, whether represented by vectors in linear codes or polynomials in cyclic codes, are

constructed so as to give a zero contribution to the error syndromes. In a linear code, codewords $\boldsymbol{c}$ satisfy $\boldsymbol{cH}^{\mathrm{T}} = 0$ and the error syndrome of a word $\boldsymbol{v}$ to be decoded is $\boldsymbol{s} = \boldsymbol{vH}^{\mathrm{T}}$ (where $\boldsymbol{H}$ is the parity-check matrix). In cyclic codes, codeword polynomials $c(x)$ satisfy $R_{g(x)}[c(x)] = 0$ and the error syndrome of a polynomial $v(x)$ to be decoded is $s(x) = R_{g(x)}[v(x)]$. With BCH codes, error syndromes are also defined, this time they are field elements in an extension field and again codewords do not contribute to the error syndromes.

Consider a t-error-correcting BCH code with $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as the roots of its generator polynomial $g(x)$. The roots of $g(x)$ are also the roots of the codeword polynomials $c(x)$, and therefore

$$c(\alpha^i) = 0 \tag{7.7}$$

for $i = 1, 2, \ldots, 2t$. Equation 7.7 provides a means for testing whether a polynomial $v(x)$ is a codeword of a BCH code. A polynomial $v(x)$ is a codeword if and only if $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ are roots of $v(x)$. Now consider a codeword $c(x)$ which incurs an error pattern $e(x)$, so giving

$$v(x) = c(x) + e(x)$$

as the polynomial to be decoded. Substituting $x = \alpha$ gives

$$v(\alpha) = c(\alpha) + e(\alpha).$$

However, from eqn 7.7, $c(\alpha) = 0$ and so

$$v(\alpha) = e(\alpha).$$

Hence $v(x)$ evaluated at $x = \alpha$ depends solely on the error pattern $e(x)$ and can therefore be used as an error syndrome of $v(x)$. Evaluating $v(x)$ at any of the field elements α^i gives

$$v(\alpha^i) = c(\alpha^i) + e(\alpha^i)$$

which again reduces to

$$v(\alpha^i) = e(\alpha^i)$$

for $i = 1, 2, \ldots, 2t$. Hence from $v(x)$ we can obtain $2t$ error syndromes and we define the ith error syndrome of $v(x)$ as

$$S_i = v(\alpha^i) \tag{7.8}$$

where $i = 1, 2, \ldots, 2t$.

The error syndromes $S_1, S_2, \ldots, S_{2t}$ are elements of the field $GF(2^m)$ containing α. Table 7.1 gives a comparison of syndrome definitions in linear, cyclic and BCH codes, along with the condition that codewords satisfy.

If $v(x)$ is error free then $v(x) = c(x)$ and

$$S_i = v(\alpha^i) = c(\alpha^i) = 0$$

Table 7.1
Error syndromes in linear, cyclic, and BCH codes

	Linear	*Cyclic*	*BCH*
Codewords	$cH^T = 0$	$R_{g(x)}[c(x)] = 0$	$c(\alpha^i) = 0$
Error syndromes	$s = vH^T$	$s(x) = R_{g(x)}[v(x)]$	$S_i = v(\alpha^i)$

for all $2t$ error syndromes. For example, consider the codeword

$$c(x) = x^9 + x^6 + x^5 + x^4 + x + 1$$

belonging to double-error-correcting (15, 7) BCH code. Let $v(x) = c(x)$ and $t = 2$, then over $GF(2^4)$ we get

$$S_1 = v(\alpha) = \alpha^9 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha + 1 = 0$$
$$S_2 = v(\alpha^2) = \alpha^3 + \alpha^{12} + \alpha^{10} + \alpha^8 + \alpha^2 + 1 = 0$$
$$S_3 = v(\alpha^3) = \alpha^{12} + \alpha^3 + 1 + \alpha^{12} + \alpha^3 + 1 = 0$$
$$S_4 = v(\alpha^4) = \alpha^6 + \alpha^9 + \alpha^5 + \alpha + \alpha^4 + 1 = 0.$$

If $v(x)$ contains errors, then some or all of the error syndromes will be nonzero. Introducing an error, say $e(x) = x^7 + x^4$, to $c(x)$ gives

$$v(x) = c(x) + e(x) = x^9 + x^7 + x^6 + x^5 + x + 1$$

and recalculating the error syndromes now gives

$$S_1 = v(\alpha) = \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1 = \alpha^3$$
$$S_2 = v(\alpha^2) = \alpha^3 + \alpha^{14} + \alpha^{12} + \alpha^{10} + \alpha^2 + 1 = \alpha^6$$
$$S_3 = v(\alpha^3) = \alpha^{12} + \alpha^6 + \alpha^3 + 1 + \alpha^3 + 1 = \alpha^4$$
$$S_4 = v(\alpha^4) = \alpha^6 + \alpha^{13} + \alpha^9 + \alpha^5 + \alpha^4 + 1 = \alpha^{12}.$$

If the number of errors does not exceed the error-correction limit of a BCH code then, as we shall see, the error pattern $e(x)$ can be determined from $S_1, S_2, \ldots, S_{2t}$.

Calculating error syndromes in a finite field can be simplified by using eqn 6.18, which shows that for $x_1, x_2, \ldots, x_n$ in $GF(2^m)$ we can write

$$x_1^2 + x_2^2 + \cdots + x_n^2 = (x_1 + x_2 + \cdots + x_n)^2.$$

For example, here the error syndrome S_2, given above can be expressed as

$$\begin{aligned} S_2 &= (\alpha^2)^9 + (\alpha^2)^7 + (\alpha^2)^6 + (\alpha^2)^5 + \alpha^2 + 1 \\ &= (\alpha^9)^2 + (\alpha^7)^2 + (\alpha^6)^2 + (\alpha^5)^2 + (\alpha)^2 + 1 \\ &= (\alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1)^2 \end{aligned}$$

and as

$$S_1 = \alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1$$

we see therefore that

$$S_2 = (\alpha^9 + \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1)^2 = S_1^2$$

Likewise $S_4 = S_2^2$ and clearly the error syndrome S_{2i} is given by

$$S_{2i} = S_i^2 \tag{7.9}$$

When calculating error syndromes we therefore need only to evaluate $S_i = v(\alpha^i)$ for odd values of i and then use $S_{2i} = S_i^2$ to obtain the error syndromes for even values of i. However, note that this only applies to binary codes, eqn 7.9 cannot be used to determine the error syndromes of nonbinary codes.

The decoder has no *a priori* knowledge of the error pattern, the only information that the decoder has is the polynomial $v(x)$ (or word $\mathbf{v}$) and the error syndromes that it calculates from $v(x)$. An error pattern with μ errors can be represented as

$$e(x) = x^{p_1} + x^{p_2} + \cdots + x^{p_\mu} \tag{7.10}$$

where the *error positions* $p_1, p_2, \ldots, p_\mu$ give the locations of the errors in the corresponding error vector $\mathbf{e}$. For example in an 8-bit word the 3-bit error pattern $\mathbf{e} = (0\,0\,1\,0\,1\,0\,0\,1)$ gives $p_1 = 5$, $p_2 = 3$, $p_3 = 0$ and so

$$e(x) = x^{p_1} + x^{p_2} + x^{p_3} = x^5 + x^3 + 1.$$

A codeword $c(x)$ incurring μ errors gives

$$v(x) = c(x) + e(x) = c(x) + (x^{p_1} + x^{p_2} + \cdots + x^{p_\mu})$$

as the word to be decoded. The decoder for a t-error correcting code evaluates $v(x)$ at $x = \alpha, \alpha^2, \ldots, \alpha^{2t}$ to obtain the error syndromes

$$\begin{aligned} S_1 &= v(\alpha) = c(\alpha) + e(\alpha) = \alpha^{p_1} + \alpha^{p_2} + \cdots + \alpha^{p_\mu} \\ S_2 &= v(\alpha^2) = c(\alpha^2) + e(\alpha^2) = \alpha^{2p_1} + \alpha^{2p_2} + \cdots + \alpha^{2p_\mu} \\ S_3 &= v(\alpha^3) = c(\alpha^3) + e(\alpha^3) = \alpha^{3p_1} + \alpha^{3p_2} + \cdots + \alpha^{3p_\mu} \\ &\vdots \\ S_{2t} &= v(\alpha^{2t}) = c(\alpha^{2t}) + e(\alpha^{2t}) = \alpha^{2tp_1} + \alpha^{2tp_2} + \cdots + \alpha^{2tp_\mu}. \end{aligned} \tag{7.11}$$

For clarity the right-hand side of eqn 7.11 is usually expressed in terms of *error-location numbers* X_i where

$$X_i = \alpha^{p_i}.$$

Note that the exponents of the error-location numbers give the error positions. The error-location numbers $X_1, X_2, \ldots, X_\mu$ are nonzero field elements in $GF(2^m)$ and provide a convenient representation of the unknown error positions $p_1, p_2, \ldots, p_\mu$.

Equation 7.11 can now be expressed as

$$\begin{aligned} S_1 &= X_1 + X_2 + X_3 + \cdots + X_\mu \\ S_2 &= X_1^2 + X_2^2 + X_3^2 + \cdots + X_\mu^2 \\ S_3 &= X_1^3 + X_2^3 + X_3^3 + \cdots + X_\mu^3 \\ &\vdots \\ S_{2t} &= X_1^{2t} + X_2^{2t} + X_3^{2t} + \cdots + X_\mu^{2t}. \end{aligned} \tag{7.12}$$

Equations 7.12 contain μ unknown variables $X_1, X_2, X_3, \ldots, X_\mu$ and $2t$ known terms $S_1, S_2, S_3, \ldots, S_{2t}$ and are referred to as the *syndrome equations*. Earlier we saw that $S_{2i} = S_i^2$ and therefore of the $2t$ syndrome equations only t equations are independent. Hence this is a set of t simultaneous equations with μ unknowns, and if $\mu \leq t$ then a unique solution exists. In other words, if the number of errors falls within the error-correction capability of the code, then the error-location numbers can be determined. The exponents of the error-location numbers are then taken as the error positions. However, there is a problem in determining the solutions of eqns 7.12. The syndrome equations are nonlinear and therefore cannot be solved using standard linear techniques such as matrix inversion. Instead, indirect methods are used involving the transformation of the syndrome equations into a form that can be readily solved. The solution of the syndrome equations lies at the heart of the decoding of BCH codes, any method that can solve the syndrome equations can be considered to be a decoding technique for the BCH codes. The most important method for decoding BCH codes is the *Peterson–Gorenstein–Zierler decoder* which is capable of dealing with multiple errors. However we first examine the simpler problems of decoding single-error-correcting and double-error-correcting BCH codes without the use of the Peterson–Gorenstein–Zierler decoder.

7.4 Decoding SEC and DEC binary BCH codes

Decoding a single-error-correcting (SEC) BCH code is quite straightforward. For a single-error-correcting code $t = 1$ and we assume that a single error (i.e. the maximum number of correctable errors) has occurred so that $\mu = 1$. Substituting $t = \mu = 1$ into the syndrome equations (eqns 7.12) gives

$$\begin{aligned} S_1 &= X_1 \\ S_2 &= X_1^2 \end{aligned} \tag{7.13}$$

and therefore the error-location number is directly given by

$$X_1 = S_1.$$

For example, consider the (7, 4) code with codeword $c(x) = x^5 + x^2 + x + 1$ and let's assume that $c(x)$ incurs the single error $e(x) = x^5$, then the word to be decoded is

$$v(x) = c(x) + e(x) = x^2 + x + 1.$$

The (7, 4) code is a single-error-correcting code constructed over $GF(2^3)$ and so the error syndromes are evaluated over $GF(2^3)$. In $GF(2^3)$

$$S_1 = v(\alpha) = \alpha^2 + \alpha + 1 = \alpha^5$$

and from the syndrome equations, given by eqns 7.13, we get $X_1 = \alpha^5$. The exponents of the error-location numbers give the error positions in the error pattern and therefore $X_1 = \alpha^5$ gives $e(x) = x^5$ as the decoder's estimate of the error pattern. The resulting codeword is

$$c(x) = v(x) + e(x) = x^5 + x^2 + x + 1$$

which we know is correct. Decoding will always be correct providing 2 or more errors do not occur. If the error syndromes are zero, then the decoder assumes that the received word is the correct codeword. Note that a syndrome table has not been used, but we have used a table for addition in $GF(2^3)$. Note also that S_2 is not required in the decoding process and therefore need not be computed.

Consider next decoding double-error-correcting (DEC) BCH codes. We again assume the occurrence of the maximum number of correctable errors, so that $\mu = t = 2$ and the syndrome equations (eqns 7.12) reduce to

$$\begin{aligned} S_1 &= X_1 + X_2 \\ S_2 &= X_1^2 + X_2^2 \\ S_3 &= X_1^3 + X_2^3 \\ S_4 &= X_1^4 + X_2^4. \end{aligned} \tag{7.14}$$

The second and fourth of these equations, involving S_2 and S_4 respectively, are dependent on the first equation and therefore solutions for X_1 and X_2 can be obtained from

$$\begin{aligned} S_1 &= X_1 + X_2 \\ S_3 &= X_1^3 + X_2^3 \end{aligned}$$

as these are two independent equations with two unknowns. The two equations are nonlinear and cannot be solved using matrix inversion, instead we proceed as follows. Consider $(X_1 + X_2)^3$:

$$\begin{aligned} (X_1 + X_2)^3 &= (X_1 + X_2)^2(X_1 + X_2) \\ &= (X_1^2 + X_2^2)(X_1 + X_2) \\ &= X_1^3 + X_2^3 + X_1X_2(X_1 + X_2). \end{aligned}$$

Substituting $X_1 + X_2 = S_1$ and $X_1^3 + X_2^3 = S_3$ into the above gives

$$S_1^3 = S_3 + S_1X_1X_2$$

and replacing X_2 by $X_2 = X_1 + S_1$ gives

$$S_1^3 = S_3 + S_1X_1(X_1 + S_1).$$

Rearranging and dividing through by S_1 gives the quadratic equation

$$X_1^2 + S_1X_1 + \frac{(S_1^3 + S_3)}{S_1} = 0 \tag{7.15}$$

and the solution of this gives X_1, with the other solution giving X_2. If we let $X_1 = X_2 + S_1$, instead of $X_2 = X_1 + S_1$, then

$$X_2^2 + S_1X_2 + \frac{(S_1^3 + S_3)}{S_1} = 0 \tag{7.16}$$

is obtained instead of eqn 7.15 and again the two roots give X_1 and X_2. Whether eqn 7.15 or 7.16 is used to determine X_1 and X_2 is quite arbitrary and we can therefore write

$$x^2 + S_1x + \frac{(S_1^3 + S_3)}{S_1} = 0 \tag{7.17}$$

where X_1 and X_2 are the two roots. The roots of eqn 7.17 can be obtained by using a Chien search, that is by systematically testing to see if field elements satisfy the equation, or by establishing the two factors (see Section 6.7). As an example, we consider the codeword

$$c(x) = x^{11} + x^8 + x^7 + x^6 + x^3 + x^2$$

belonging to the double-error-correcting (15, 7) BCH code. Introducing an error pattern, say $e(x) = x^{10} + x^2$ gives

$$v(x) = x^{11} + x^{10} + x^8 + x^7 + x^6 + x^3$$

and over $GF(2^4)$ the error syndromes S_1 and S_3 are

$$S_1 = v(\alpha) = \alpha^{11} + \alpha^{10} + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^3 = \alpha^4$$

$$S_3 = v(\alpha^3) = \alpha^3 + 1 + \alpha^9 + \alpha^6 + \alpha^3 + \alpha^9 = \alpha^{13}.$$

Evaluating $(S_1^3 + S_3)/S_1$ gives

$$\frac{(S_1^3 + S_3)}{S_1} = \frac{(\alpha^{12} + \alpha^{13})}{\alpha^4} = \frac{\alpha}{\alpha^4} = \alpha^{12}$$

and substituting this into eqn 7.17, along with $S_1 = \alpha^4$, gives

$$x^2 + \alpha^4x + \alpha^{12} = 0.$$

The two roots of this quadratic equation give the required error-location numbers. Let $p(x) = x^2 + \alpha^4x + \alpha^{12}$, then using a Chien search we systematically test the

nonzero field elements of $GF(2^4)$ to see if they are roots of $p(x)$. Starting with $x = 1$

$$p(1) = \alpha^{13}$$
$$p(\alpha) = \alpha^{13}$$
$$p(\alpha^2) = 0$$
$$p(\alpha^3) = \alpha^3.$$

So far α^2 is one solution. We could continue searching for the second root but it is easier to use the first expression in eqns 7.14, namely $S_1 = X_1 + X_2$, which gives

$$X_2 = S_1 + X_1 = \alpha^4 + \alpha^2 = \alpha^{10}$$

as the other solution. The reader can verify that $p(\alpha^{10}) = 0$. The error-location numbers are therefore $X_1 = \alpha^2$ and $X_2 = \alpha^{10}$, the exponents of the field elements X_1 and X_2 correspond to the errors x^2 and x^{10} respectively. We have therefore correctly determined the error polynomial $e(x) = x^{10} + x^2$ present in $v(x)$.

In the event of a single error occurring there will be only one nonzero error-location number and so $X_1 = S_1$, $X_2 = 0$, $S_3 = X_1^3 = S_1^3$ and eqn 7.17 reduces to

$$x + S_1 = 0.$$

The error-location number is therefore directly given by S_1 (i.e. it is the same as decoding a single-error-correcting code). For example consider again the (15, 7) double-error-correcting BCH code with codeword $c(x) = x^{11} + x^8 + x^7 + x^6 + x^3 + x^2$, but this time incurring the single error $e(x) = x^4$. Here $v(x) = x^{11} + x^8 + x^7 + x^6 + x^4 + x^3 + x^2$ giving the error syndromes

$$S_1 = v(\alpha) = \alpha^4$$
$$S_3 = v(\alpha^3) = \alpha^{12}$$

over $GF(2^4)$. Evaluating $(S_1^3 + S_3)/S_1$ gives

$$\frac{(S_1^3 + S_3)}{S_1} = \frac{(\alpha^4)^3 + \alpha^{12}}{\alpha^4} = \frac{\alpha^{12} + \alpha^{12}}{\alpha^4} = 0.$$

The constant term in eqn 7.17 is therefore 0 and so eqn 7.17 reduces to

$$x + S_1 = 0$$

as required. Therefore the error-location number $X = S_1 = \alpha^4$ which gives the error pattern x^4 (which we know is correct).

Example 7.2

Given that the codewords $c_1(x)$ and $c_2(x)$, belonging to the double-error-correcting (15, 7) code constructed over $GF(2^4)$, incur 2 and 1 errors so giving

(a) $v_1(x) = x^{11} + x^9 + x^8 + x^6 + x^5 + x + 1$

(b) $v_2(x) = x^{12} + x^{11} + x^{10} + x^9 + x^7 + x^5 + x$

respectively, determine $c_1(x)$ and $c_2(x)$.

(a) The error syndromes are

$$S_1 = v_1(\alpha) = \alpha^3$$
$$S_3 = v_1(\alpha^3) = \alpha^{13}$$

over $GF(2^4)$. Substituting these into eqn 7.17 gives

$$x^2 + \alpha^3 x + \alpha^7 = 0.$$

By inspection we can see that $\alpha^{10} + \alpha^{12} = \alpha^3$ and $\alpha^{10}\alpha^{12} = \alpha^7$ over $GF(2^4)$ and so

$$x^2 + \alpha^3 x + \alpha^7 = (x + \alpha^{10})(x + \alpha^{12}) = 0.$$

The roots of $x^2 + \alpha^3 x + \alpha^7 = 0$ are therefore α^{10} and α^{12}, giving the error-location numbers $X_1 = \alpha^{10}$ and $X_2 = \alpha^{12}$. Hence

$$e(x) = x^{10} + x^{12}$$

and so

$$c_1(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x + 1.$$

(b) Here we have $S_1 = \alpha^4$ and $S_3 = \alpha^{12}$. Therefore $S_1^3 + S_3 = \alpha^{12} + \alpha^{12} = 0$ and eqn 7.17 reduces to $x + \alpha^4 = 0$. The error-location number is $X_1 = \alpha^4$ giving an error pattern $e(x) = x^4$ and codeword $c_2(x) = x^{12} + x^{11} + x^{10} + x^9 + x^7 + x^5 + x^4 + x$. □

To summarize, when decoding a double-error-correcting BCH code the occurrence of two errors results in a quadratic equation whose two roots give two error-location numbers. However, in the event of a single error occurring, the quadratic equation reduces to a linear equation and the error-location number is given by S_1 (as for a single-error-correcting code). In the event of three or more errors occurring a decoding error will occur if eqn 7.17 has 1 or 2 solutions. If eqn 7.17 has no solution then an uncorrectable error pattern will have been detected. If the error pattern is identical to a codeword, then the syndromes are zero and again a decoding error occurs.

Example 7.3

Consider the (15, 7) double-error-correcting BCH code and codeword $c(x) = x^8 + x^7 + x^6 + x^4 + 1$. Determine the outcome of a decoder when $c(x)$ incurs the error patterns

(a) $e(x) = x^7 + x^2 + 1$

(b) $e(x) = x^{11} + x^9 + x^6 + x^4$.

(a) The polynomial to be decoded is

$$v(x) = c(x) + e(x) = x^8 + x^6 + x^4 + x^2$$

giving error syndromes

$$S_1 = v(\alpha) = \alpha^8 + \alpha^6 + \alpha^4 + \alpha^2 = \alpha^{11}$$
$$S_3 = v(\alpha^3) = \alpha^9 + \alpha^3 + \alpha^{12} + \alpha^6 = 1$$

over $GF(2^4)$. Substituting S_1 and S_3 in eqn 7.17 gives

$$x^2 + \alpha^{11} x + \alpha^3 = 0.$$

It can be shown, by inspection or by testing the field elements, that none of the elements in $GF(2^4)$ are solutions of $x^2 + \alpha^{11} x + \alpha^3 = 0$. No error-location numbers can therefore be obtained and the decoder concludes that an uncorrectable error pattern has been detected, i.e. a decoding failure occurs.

(b) Here $v(x) = x^{11} + x^9 + x^8 + x^7 + 1$ and so

$$S_1 = v(\alpha) = \alpha^7$$
$$S_3 = v(\alpha^3) = 0.$$

Substituting S_1 and S_3 into eqn 7.17 gives

$$x^2 + \alpha^7 x + \alpha^{14} = 0$$

and by inspection it can be seen that $\alpha^2 + \alpha^{12} = \alpha^7$ and $\alpha^2\alpha^{12} = \alpha^{14}$ over $GF(2^4)$. Hence

$$x^2 + \alpha^7 x + \alpha^{14} = (x + \alpha^2)(x + \alpha^{12})$$

giving α^2 and α^{12} as the required roots and error-location numbers. The decoder therefore concludes that the double error pattern $e(x) = x^2 + x^{12}$ occurred and adding this to $v(x)$ gives the codeword

$$c(x) = x^{12} + x^{11} + x^9 + x^8 + x^7 + x^2 + 1.$$

This is the wrong codeword and so a decoding error has occurred. □

7.5 The error-location polynomial

The method described in Section 7.4 for decoding single error-correcting and double-error-correcting BCH codes can be extended to deal with multiple-error-correcting BCH codes. For a t-error-correcting code a polynomial of degree t or less can be defined whose coefficients are functions of the error syndromes. The occurrence of $\mu \leq t$ errors gives a polynomial of degree μ whose μ roots are the reciprocal of the required error-location numbers. Consider again eqn 7.17 and let

$$\sigma_1 = S_1 \qquad (7.18)$$
$$\sigma_2 = (S_1^3 + S_3)/S_1$$

then eqn 7.17 becomes

$$x^2 + \sigma_1 x + \sigma_2 = 0.$$

This is the polynomial of highest degree that we need to consider when decoding double-error-correcting codes. In the event of 1 error occurring we get $\sigma_2 = 0$ and therefore $x + \sigma_1 = 0$, which gives $X = \sigma_1 = S_1$ as the required error-location number. For a t-error-correcting code we need to consider polynomials of the form

$$x^\mu + \sigma_1 x^{\mu-1} + \sigma_2 x^{\mu-2} + \cdots + \sigma_{\mu-1} x + \sigma_\mu = 0$$

where $\mu \leq t$ and where the polynomial coefficients are again functions of the error syndromes and the μ roots give the μ error-location numbers. If we replace x by its

reciprocal $1/x$ and then multiply through by x^{μ} we get

$$\sigma_0 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_{\mu-1}x^{\mu-1} + \sigma_\mu x^\mu = 0$$

where $\sigma_0 = 1$. We now define the *error-location polynomial*

$$\sigma(x) = \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_{\mu-1}x^{\mu-1} + \sigma_\mu x^\mu \tag{7.19}$$

which is a polynomial whose roots are the reciprocal of the error-location numbers. The error-location polynomial can be defined so that its roots are error-location numbers, and whilst it is easier to think of roots as representing error-location numbers, it is however convenient and conventional to use error-location polynomials whose reciprocal roots are error-location numbers. Note that for a double-error-correcting code the error-location polynomial is

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 \tag{7.20}$$

where σ_1 and σ_2 are again given by eqns 7.18.

Example 7.4
Given that $v(x) = x^9 + x^8 + x^6 + x^4 + 1$ represents a codeword $c(x)$, of the double-error-correcting (15, 7) code, that has incurred 2 errors determine $c(x)$.

Over $GF(2^4)$ we get

$$S_1 = v(\alpha) = 1$$
$$S_3 = v(\alpha^3) = \alpha^4$$

and substituting these in to eqns 7.18 gives

$$\sigma_1 = 1$$
$$\sigma_2 = \alpha$$

The error-location polynomial is therefore

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 = 1 + x + \alpha x^2$$

Using a Chien search we find that the roots of $\sigma(x)$ are α^8 and α^4 over $GF(2^4)$. The error-location numbers are therefore

$$X_1 = 1/\alpha^8 = \alpha^9$$
$$X_2 = 1/\alpha^4 = \alpha^7$$

which give the error polynomial $e(x) = x^9 + x^7$ and codeword polynomial

$$c(x) = v(x) + e(x) = x^8 + x^7 + x^6 + x^4 + 1 \qquad \square$$

If the roots of the error-location polynomial are the field elements $\beta_1, \beta_2, \ldots, \beta_\mu$ then the error-location numbers are

$$X_1 = 1/\beta_1$$
$$X_2 = 1/\beta_2$$
$$\vdots$$
$$X_\mu = 1/\beta_\mu$$

and the error-location polynomial can be expressed as

$$\sigma(x) = (xX_1 + 1)(xX_2 + 1)\cdots(xX_\mu + 1) \tag{7.21}$$

Determining the error-location polynomial is the most difficult part of decoding a BCH code. The coefficients $\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_\mu$ of the error-location polynomial $\sigma(x)$, have to be determined from the known error syndromes. To achieve this eqn 7.21 is expanded and its coefficients are compared with those of eqn 7.19, and in doing so we find that

$$\begin{aligned}
\sigma_0 &= 1\\
\sigma_1 &= X_1 + X_2 + X_3 + \cdots + X_{\mu-1} + X_\mu\\
\sigma_2 &= X_1X_2 + X_2X_3 + X_3X_4 + \cdots + X_{\mu-1}X_\mu\\
&\vdots\\
\sigma_\mu &= X_1X_2X_3\cdots X_{\mu-1}X_\mu
\end{aligned} \tag{7.22}$$

The coefficients of $\sigma(x)$ as given above are said to be *elementary symmetric functions* of the error-location numbers. We now have two sets of equations involving the error-location numbers of a t-error-correcting code:

(1) equations 7.12 relating the error-location numbers to the error syndromes;
(2) equations 7.22 relating the error-location numbers to the polynomial coefficients.

From these two sets of equations we can eliminate the error-location numbers to obtain expressions involving only the error syndromes and the coefficients of the error-location polynomial. It can be shown that for the first μ error syndromes

$$\begin{aligned}
S_1 &= \sigma_1\\
S_2 &= \sigma_1 S_1 + 2\sigma_2\\
S_3 &= \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3\\
&\vdots\\
S_\mu &= \sigma_1 S_{\mu-1} + \sigma_2 S_{\mu-2} + \cdots + \sigma_{\mu-1}S_1 + \mu\sigma_\mu
\end{aligned} \tag{7.23}$$

Note that the last term $i\sigma_i$ in each expression in the eqns 7.23 is 0 for even values of i and σ_i for odd values of i. The remaining error syndromes are given by

$$\begin{aligned}
S_{\mu+1} &= \sigma_1 S_\mu + \sigma_2 S_{\mu-1} + \cdots + \sigma_{\mu-1}S_2 + \sigma_\mu S_1\\
S_{\mu+2} &= \sigma_1 S_{\mu+1} + \sigma_2 S_\mu + \cdots + \sigma_{\mu-1}S_3 + \sigma_\mu S_2\\
S_{\mu+3} &= \sigma_1 S_{\mu+2} + \sigma_2 S_{\mu+1} + \cdots + \sigma_{\mu-1}S_4 + \sigma_\mu S_3\\
&\vdots\\
S_{2\mu} &= \sigma_1 S_{2\mu-1} + \sigma_2 S_{2\mu-2} + \cdots + \sigma_{\mu-1}S_{\mu+1} + \sigma_\mu S_\mu
\end{aligned} \tag{7.24}$$

Equations 7.23 and 7.24 are a set of linear equations, referred to as *Newton's identities*, from which the coefficients of $\sigma(x)$ can be determined. Although they are a single set of equations, they fall naturally into two groups, the first μ equations given by eqns 7.23, and the remaining μ equations given by eqns 7.24. For clarity, they can

be expressed in the matrix forms

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_\mu \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ S_1 & 2 & 0 & \dots & 0 & 0 \\ S_2 & S_1 & 3 & \dots & 0 & 0 \\ \vdots & & & & & \vdots \\ S_{\mu-1} & S_{\mu-2} & S_{\mu-3} & \dots & S_1 & \mu \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_\mu \end{bmatrix} \tag{7.25}$$

and

$$\begin{bmatrix} S_{\mu+1} \\ S_{\mu+2} \\ S_{\mu+3} \\ \vdots \\ S_{2\mu} \end{bmatrix} = \begin{bmatrix} S_\mu & S_{\mu-1} & S_{\mu-2} & \dots & S_2 & S_1 \\ S_{\mu+1} & S_\mu & S_{\mu-1} & \dots & S_3 & S_2 \\ S_{\mu+2} & S_{\mu+1} & S_\mu & \dots & S_4 & S_3 \\ \vdots & & & & & \vdots \\ S_{2\mu-1} & S_{2\mu-2} & S_{2\mu-3} & \dots & S_{\mu+1} & S_\mu \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_\mu \end{bmatrix} \tag{7.26}$$

In Section 7.6 we look at the Peterson–Gorenstein–Zierler decoder, which uses eqn 7.26 as the basis for a decoder for multiple-error-correcting BCH codes. Later Berlekamp's algorithm is considered, this is a fast algorithm that uses eqns 7.23 and 7.24. For the remaining part of this section, we look at how the coefficients of the error-location polynomial of a binary code can be obtained algebraically from eqns 7.23–7.26.

For a binary code eqns 7.23–7.26 can be simplified by taking into account the relationship $S_{2i} = S_i^2$. Consider eqns 7.23, using $S_2 = S_1^2$ we find that the second equation $S_2 = \sigma_1 S_1 + 2\sigma_2$ reduces to the first equation $S_1 = \sigma_1$ and can therefore be excluded. Likewise all the equations for S_i with even values of i can be excluded from eqns 7.23–7.26. Furthermore we can combine eqns 7.25 and 7.26 to get

$$\begin{bmatrix} S_1 \\ S_3 \\ S_5 \\ \vdots \\ S_{2\mu-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ S_2 & S_1 & 1 & \dots & 0 & 0 \\ S_4 & S_3 & S_2 & \dots & 0 & 0 \\ \vdots & & & & & \vdots \\ S_{2\mu-2} & S_{2\mu-3} & S_{2\mu-4} & \dots & S_\mu & S_{\mu-1} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_\mu \end{bmatrix} \tag{7.27}$$

For a double-error-correcting code the maximum number of correctable errors is 2, and setting $\mu = 2$ in eqn 7.27 gives

$$\begin{bmatrix} S_1 \\ S_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ S_2 & S_1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix}$$

from which we get

$$S_1 = \sigma_1$$
$$S_3 = S_2\sigma_1 + S_1\sigma_2.$$

Rearranging these 2 expressions gives

$$\sigma_1 = S_1$$
$$\sigma_2 = (S_1^3 + S_3)/S_1$$

which agree with the coefficients of $\sigma(x)$ given previously (eqns 7.18).

Next we consider the slightly more difficult example of a triple-error-correcting code. Here the error-location polynomial is

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \sigma_3 x^3$$

and as the maximum number of correctable errors is $\mu = 3$ eqn 7.27 reduces to

$$\begin{bmatrix} S_1 \\ S_3 \\ S_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ S_2 & S_1 & 1 \\ S_4 & S_3 & S_2 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix}$$

which when expanded gives

$$\begin{aligned} S_1 &= \sigma_1 \\ S_3 &= S_2\sigma_1 + S_1\sigma_2 + \sigma_3 \\ S_5 &= S_4\sigma_1 + S_3\sigma_2 + S_2\sigma_3. \end{aligned} \tag{7.28}$$

Multiplying the middle equation by S_2 and adding it to the last equation eliminates σ_1 and σ_3, so allowing σ_2 to be determined

$$\begin{aligned} S_2S_3 + S_5 &= (S_2^2 + S_4)\sigma_1 + (S_2S_1 + S_3)\sigma_2 + (S_2 + S_2)\sigma_3 \\ &= (S_1^4 + S_1^4)\sigma_1 + (S_1^2S_1 + S_3)\sigma_2 \\ &= (S_1^3 + S_3)\sigma_2 \end{aligned}$$

and so

$$\sigma_2 = \frac{(S_2S_3 + S_5)}{(S_1^3 + S_3)}.$$

Now rearranging the middle expression of eqn 7.28 gives

$$\sigma_3 = S_3 + S_2\sigma_1 + S_1\sigma_2$$

and we could leave σ_3 like this since σ_1 and σ_2 are known. However, for completeness, we can substitute σ_1 and σ_2 as given above into σ_3 to get

$$\sigma_3 = (S_3 + S_1^3) + \frac{S_1(S_2S_3 + S_5)}{(S_1^3 + S_3)}.$$

Therefore for a triple-error-correcting code the coefficients of the error-location polynomial are

$$\begin{aligned} \sigma_1 &= S_1 \\ \sigma_2 &= \frac{(S_2S_3 + S_5)}{(S_1^3 + S_3)} \\ \sigma_3 &= (S_3 + S_1^3) + \frac{S_1(S_2S_3 + S_5)}{(S_1^3 + S_3)}. \end{aligned} \tag{7.29}$$

Example 7.5
Given a triple-error-correcting code and error syndromes $S_1 = \alpha^3$, $S_3 = \alpha^8$ and $S_5 = 1$ over $GF(2^4)$, determine the error-location polynomial $\sigma(x)$.

Using eqns 7.29 gives

$$\sigma_1 = S_1 = \alpha^3$$
$$\sigma_2 = \frac{(S_3 S_1 + S_5)}{(S_1^3 + S_3)} = \frac{(\alpha^6 \alpha^8 + 1)}{(\alpha^9 + \alpha^8)} = \frac{\alpha^3}{\alpha^{12}} = \alpha^6$$
$$\sigma_3 = (S_1 + S_1^3) + \frac{S_1(S_2 S_3 + S_5)}{(S_1^3 + S_3)} = (\alpha^9 + \alpha^8) + \frac{(\alpha^9 \alpha^8 + \alpha^3 1)}{(\alpha^9 + \alpha^8)} = \alpha^8$$

The error-location-polynomial is therefore

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \sigma_3 x^3 = 1 + \alpha^3 x + \alpha^6 x^2 + \alpha^8 x^3. \qquad \square$$

The method described in this section can be applied to any t-error-correcting code. However beyond $t = 4$ or 5 the resulting equations, relating the coefficients of the error-location polynomial to the error syndromes, become rather complicated and so this approach becomes impractical. Instead the Peterson–Gorenstein–Zierler decoder forms the basis for multiple-error correction.

7.6 The Peterson–Gorenstein–Zierler decoder

We are now in a position where we can consider an algorithm that typifies the decoding of BCH codes, namely the *Peterson–Gorenstein–Zierler decoder*. This is a general purpose decoder that can be used for decoding any t-error-correcting BCH code. It is based on the error-location polynomial, and as we shall see the decoder brings together into a single algorithm the various ideas considered in the previous sections.

The error syndromes $S_{\mu+1}, S_{\mu+2}, \ldots, S_{2\mu}$ are related to the coefficients of the error-location polynomial by eqn 7.26. By convention the order of the columns of the matrix in eqn 7.26 are reversed, along with the rows of the column vector containing the polynomial coefficients. This gives

$$\begin{bmatrix} S_{\mu+1} \\ S_{\mu+2} \\ S_{\mu+3} \\ \vdots \\ S_{2\mu} \end{bmatrix} = \begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{\mu-1} & S_\mu \\ S_2 & S_3 & S_4 & \cdots & S_\mu & S_{\mu+1} \\ S_3 & S_4 & S_5 & \cdots & S_{\mu+1} & S_{\mu+2} \\ \vdots & & & & & \vdots \\ S_\mu & S_{\mu+1} & S_{\mu+2} & \cdots & S_{2\mu-2} & S_{2\mu-1} \end{bmatrix} \begin{bmatrix} \sigma_\mu \\ \sigma_{\mu-1} \\ \sigma_{\mu-2} \\ \vdots \\ \sigma_1 \end{bmatrix} \qquad (7.30)$$

which can be expressed as

$$\boldsymbol{S} = \boldsymbol{M}\boldsymbol{\sigma}$$

where

$$\boldsymbol{M} = \begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{\mu-1} & S_\mu \\ S_2 & S_3 & S_4 & \cdots & S_\mu & S_{\mu+1} \\ S_3 & S_4 & S_5 & \cdots & S_{\mu+1} & S_{\mu+2} \\ \vdots & & & & & \vdots \\ S_\mu & S_{\mu+1} & S_{\mu+2} & \cdots & S_{2\mu-2} & S_{2\mu-1} \end{bmatrix} \qquad (7.31)$$

$$\boldsymbol{S} = \begin{bmatrix} S_{\mu+1} \\ S_{\mu+2} \\ S_{\mu+3} \\ \vdots \\ S_{2\mu} \end{bmatrix} \qquad (7.32)$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_\mu \\ \sigma_{\mu-1} \\ \sigma_{\mu-2} \\ \vdots \\ \sigma_1 \end{bmatrix} \qquad (7.33)$$

Assuming that $\boldsymbol{M}$ is nonsingular, so that its inverse $\boldsymbol{M}^{-1}$ exists, and multiplying $\boldsymbol{S} = \boldsymbol{M}\boldsymbol{\sigma}$ through by $\boldsymbol{M}^{-1}$ gives

$$\boldsymbol{M}^{-1}\boldsymbol{S} = (\boldsymbol{M}^{-1}\boldsymbol{M})\boldsymbol{\sigma} = \boldsymbol{\sigma}$$

and therefore the coefficients of the error-location polynomial are given by

$$\boldsymbol{\sigma} = \boldsymbol{M}^{-1}\boldsymbol{S}. \qquad (7.34)$$

To evaluate eqn 7.34, $\boldsymbol{M}$ has to be nonsingular and we need to know the number of errors μ that have occurred. It can be shown that the matrix

$$\boldsymbol{M} = \begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{i-1} & S_i \\ S_2 & S_3 & S_4 & \cdots & S_i & S_{i+1} \\ S_3 & S_4 & S_5 & \cdots & S_{i+1} & S_{i+2} \\ \vdots & & & & & \vdots \\ S_i & S_{i+1} & S_{i+2} & \cdots & S_{2i-2} & S_{2i-1} \end{bmatrix}$$

is nonsingular if $i = \mu$, but singular if $i > \mu$. For example consider a code that can correct 5 errors but only 3 errors actually occur, so $t = 5$ and $\mu = 3$. Taking $i = 5$ and constructing

$$\boldsymbol{M} = \begin{bmatrix} S_1 & S_2 & S_3 & S_4 & S_5 \\ S_2 & S_3 & S_4 & S_5 & S_6 \\ S_3 & S_4 & S_5 & S_6 & S_7 \\ S_4 & S_5 & S_6 & S_7 & S_8 \\ S_5 & S_6 & S_7 & S_8 & S_9 \end{bmatrix}$$

will give $\det(\boldsymbol{M}) = 0$, and so $\boldsymbol{M}$ is singular. Likewise the 4 by 4 matrix $\boldsymbol{M}$ of error syndromes, obtained when $i = 4$ will also be singular. However, when i equals the number of errors that have occurred, i.e. $i = \mu = 3$, the 3 by 3 matrix $\boldsymbol{M}$ is non-singular. $\boldsymbol{M}^{-1}$ can then be determined and the polynomial coefficients can be found using $\boldsymbol{\sigma} = \boldsymbol{M}^{-1}\boldsymbol{S}$.

For a BCH code with error-correction limit t, decoding proceeds as follows. The decoder first assumes that the maximum number of correctable errors have occurred, $i = t$, and constructs $\boldsymbol{M}$ and determines $\det(\boldsymbol{M})$. If $\boldsymbol{M}$ is nonsingular, then $\boldsymbol{M}^{-1}$ and $\boldsymbol{\sigma} = \boldsymbol{M}^{-1}\boldsymbol{S}$ can be found. If $\boldsymbol{M}$ is singular the decoder assumes that t errors did not occur and repeats the calculations on the assumption of 1 less error, i.e. $i = t - 1$. The decoder continues in an iterative manner, decreasing i by 1 each time $\boldsymbol{M}$ is found to be singular. On obtaining $\det(\boldsymbol{M}) \neq 0$, the value of i is taken to be the number of errors that occurred and $\boldsymbol{\sigma}$ is determined. The main steps of the Peterson–Gorenstein–Zierler decoder are

1. Calculate the error syndromes $S_1, S_2, \ldots, S_{2t}$ from $v(x)$.
2. Assume the maximum number of errors, $i = t$.
3. Construct the matrix $\boldsymbol{M}$.
4. Find the determinant of $\boldsymbol{M}$ and check if $\det(\boldsymbol{M}) = 0$. If $\det(\boldsymbol{M}) = 0$ reduce i by 1 and go back to step 3, otherwise continue to Step 5.
5. Determine $\boldsymbol{M}^{-1}$ and construct $\boldsymbol{S}$.
6. Find the polynomial coefficients using $\boldsymbol{\sigma} = \boldsymbol{M}^{-1}\boldsymbol{S}$ and construct $\sigma(x)$ from $\boldsymbol{\sigma}$.
7. Determine the roots of $\sigma(x)$ and take their reciprocals. The error-location numbers are given by the reciprocal roots.

As an example of how the decoder works, let's consider the triple-error-correcting (15, 5) BCH code with

$$v(x) = x^8 + x^5 + x^2 + x + 1$$

where $v(x)$ corresponds to a codeword $c(x)$ with 2 errors. For clarity, the example is referenced to the seven steps given above.

Step 1 In $GF(2^4)$ the error syndromes are

$$\begin{aligned}
S_1 &= \alpha^8 + \alpha^5 + \alpha^2 + \alpha + 1 = \alpha^2 \\
S_2 &= S_1^2 = \alpha^4 \\
S_3 &= \alpha^9 + \alpha^{15} + \alpha^6 + \alpha^3 + 1 = \alpha^{11} \\
S_4 &= S_2^2 = \alpha^8 \\
S_5 &= \alpha^{40} + \alpha^{25} + \alpha^{10} + \alpha^5 + 1 = 0 \\
S_6 &= S_3^2 = \alpha^7
\end{aligned}$$

Step 2 Assume the maximum number of errors, $i = 3$.

Step 3 The matrix $\boldsymbol{M}$ is

$$\boldsymbol{M} = \begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha^4 & \alpha^{11} \\ \alpha^4 & \alpha^{11} & \alpha^8 \\ \alpha^{11} & \alpha^8 & 0 \end{bmatrix}$$

Step 4 The determinant of $\boldsymbol{M}$ is

$$\begin{aligned}
\det(\boldsymbol{M}) &= \begin{vmatrix} \alpha^2 & \alpha^4 & \alpha^{11} \\ \alpha^4 & \alpha^{11} & \alpha^8 \\ \alpha^{11} & \alpha^8 & 0 \end{vmatrix} \\
&= \alpha^2 \begin{vmatrix} \alpha^{11} & \alpha^8 \\ \alpha^8 & 0 \end{vmatrix} + \alpha^4 \begin{vmatrix} \alpha^4 & \alpha^8 \\ \alpha^{11} & 0 \end{vmatrix} + \alpha^{11} \begin{vmatrix} \alpha^4 & \alpha^{11} \\ \alpha^{11} & \alpha^8 \end{vmatrix} \\
&= \alpha^2\alpha + \alpha^4\alpha^4 + \alpha^{11}\alpha^2 = \alpha^3 + \alpha^8 + \alpha^{13} = 0
\end{aligned}$$

The matrix $\boldsymbol{M}$ is therefore singular, so i is reduced by 1 to give $i = 2$ and steps 3 and 4 are repeated.

Step 3—repeated We now have

$$\boldsymbol{M} = \begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha^4 \\ \alpha^4 & \alpha^{11} \end{bmatrix}$$

Step 4—repeated The revised value of $\det(\boldsymbol{M})$ is

$$\det(\boldsymbol{M}) = \begin{vmatrix} \alpha^2 & \alpha^4 \\ \alpha^4 & \alpha^{11} \end{vmatrix} = \alpha^{13} + \alpha^8 = \alpha^3.$$

Now $\boldsymbol{M}$ is nonsingular so we can move to Step 5 and find $\boldsymbol{M}^{-1}$.

Step 5 Using $\boldsymbol{M}^{-1} = \text{adj}\,(\boldsymbol{M})/\det(\boldsymbol{M})$ we get

$$\text{adj}(\boldsymbol{M}) = \begin{bmatrix} \alpha^{11} & \alpha^4 \\ \alpha^4 & \alpha^2 \end{bmatrix}$$

and so

$$\boldsymbol{M}^{-1} = \frac{1}{\alpha^3}\begin{bmatrix} \alpha^{11} & \alpha^4 \\ \alpha^4 & \alpha^2 \end{bmatrix} = \begin{bmatrix} \alpha^8 & \alpha \\ \alpha & \alpha^{14} \end{bmatrix}$$

Also

$$\boldsymbol{S} = \begin{bmatrix} S_3 \\ S_4 \end{bmatrix} = \begin{bmatrix} \alpha^{11} \\ \alpha^8 \end{bmatrix}.$$

Step 6 Using $\boldsymbol{\sigma} = \boldsymbol{M}^{-1}\boldsymbol{S}$ gives

$$\begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha^8 & \alpha \\ \alpha & \alpha^{14} \end{bmatrix}\begin{bmatrix} \alpha^{11} \\ \alpha^8 \end{bmatrix} = \begin{bmatrix} \alpha^{14} \\ \alpha^2 \end{bmatrix}$$

and so $\sigma_2 = \alpha^{14}$, $\sigma_1 = \alpha^2$ and the error-location polynomial is therefore

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 = 1 + \alpha^2 x + \alpha^{14} x^2$$

Step 7 Using a Chien over $GF(2^4)$ we find that the roots of $\sigma(x)$ are α^5 and α^{11}, the reciprocals of which give the error-location numbers α^{10} and α^4 respectively. The error pattern is therefore $x^{10}+x^4$ and so the required codeword polynomial is

$$c(x) = v(x) + x^{10} + x^4 = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1.$$

In the example below the (15, 5) triple-error-correcting BCH code is again considered, but with a codeword that has incurred 3 errors.

Example 7.6
A codeword $c(x)$, of the (15, 5) triple-error-correcting BCH code, incurs errors so as to give $v(x) = x^{13} + x^{10} + x^8 + x^4 + x + 1$. Find the number of errors that $c(x)$ has incurred, the error pattern and $c(x)$.

Over $GF(2^4)$ the error syndromes are

$$\begin{aligned}
S_1 &= v(\alpha) = \alpha^{13} + \alpha^{10} + \alpha^8 + \alpha^4 + \alpha + 1 = \alpha^{12}\\
S_2 &= S_1^2 = \alpha^9\\
S_3 &= v(\alpha^3) = \alpha^9 + 1 + \alpha^9 + \alpha^{12} + \alpha^3 + 1 = \alpha^{10}\\
S_4 &= S_2^2 = \alpha^3\\
S_5 &= \alpha^5 + \alpha^5 + \alpha^{10} + \alpha^5 + \alpha^5 + 1 = \alpha^5\\
S_6 &= S_3^2 = \alpha^5.
\end{aligned}$$

Assuming the maximum number of errors, $i = 3$, we construct the matrix

$$\boldsymbol{M} = \begin{bmatrix} \alpha^{12} & \alpha^9 & \alpha^{10} \\ \alpha^9 & \alpha^{10} & \alpha^3 \\ \alpha^{10} & \alpha^3 & \alpha^5 \end{bmatrix}.$$

The determinant of $\boldsymbol{M}$ is

$$\det(\boldsymbol{M}) = \alpha^{12}(1+\alpha^6) + \alpha^9(\alpha^{14} + \alpha^{13}) + \alpha^{10}(\alpha^{12} + \alpha^5) = \alpha^4.$$

As $\boldsymbol{M}$ is nonsingular we assume that 3 errors have occurred. The inverse of M is

$$\boldsymbol{M}^{-1} = \frac{\text{adj}(\boldsymbol{M})}{\det(\boldsymbol{M})} = (1/\alpha^4)\begin{bmatrix} \alpha^{13} & \alpha^2 & \alpha^{14} \\ \alpha^2 & \alpha & \alpha \\ \alpha^{14} & \alpha & \alpha^4 \end{bmatrix} = \begin{bmatrix} \alpha^9 & \alpha^{13} & \alpha^{10} \\ \alpha^{13} & \alpha^{12} & \alpha^{12} \\ \alpha^{10} & \alpha^{12} & 1 \end{bmatrix}$$

and

$$\boldsymbol{S} = \begin{bmatrix} \alpha^3 \\ \alpha^5 \\ \alpha^5 \end{bmatrix}.$$

Using $\sigma = \boldsymbol{M}^{-1}\boldsymbol{S}$ gives

$$\begin{bmatrix} \sigma_3 \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha^9 & \alpha^{13} & \alpha^{10} \\ \alpha^{13} & \alpha^{12} & \alpha^{12} \\ \alpha^{10} & \alpha^{12} & 1 \end{bmatrix}\begin{bmatrix} \alpha^3 \\ \alpha^5 \\ \alpha^5 \end{bmatrix} = \begin{bmatrix} \alpha^5 \\ \alpha \\ \alpha^{12} \end{bmatrix}$$

and so $\sigma_3 = \alpha^5$, $\sigma_2 = \alpha$, $\sigma_1 = \alpha^{12}$ giving the error-location polynomial

$$\sigma(x) = 1 + \alpha^{12}x + \alpha x^2 + \alpha^5 x^3.$$

Searching $GF(2^4)$ for the roots of $\sigma(x)$ gives α^2, α^{10}, and α^{13} as roots and taking their reciprocals gives α^{13}, α^5, and α^2 as the error-location numbers respectively. The error pattern is therefore $e(x) = x^{13} + x^5 + x^2$ and the codeword polynomial $c(x) = v(x) + e(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$. □

The Peterson–Gorenstein–Zierler decoder forms the basis of decoding algorithms for BCH codes. It is a relatively simple decoder, as the reader should find after working through a few examples. The matrix inversion does however present a problem to the decoder. For a large error-correction limit t, evaluating the resulting determinants can be computationally slow and inefficient. Furthermore, as we shall see later, a second matrix inversion is required when dealing with non-binary codes, and so aggravating the problem. To develop fast decoding algorithms we need to avoid the matrix inversions, this is considered in Sections 7.8 and 7.9.

7.7 Reed–Solomon codes

The codes considered so far have all been binary codes, and we now turn our attention to non-binary codes. At first the idea of a non-binary code may seem rather strange or of little practical use. Information processing, transmission, and storage is usually thought of in terms of a binary representation. Bits are manipulated either individually or in blocks of convenient length, for example as 8-bit words. However, an 8-bit word can be thought of as a single *non-binary symbol* with 256 different values, irrespective of its underlying structure (i.e. the fact that it is really a collection of 8 bits and not a single symbol). Likewise any sequence of r bits can be viewed as a single non-binary symbol that has one of 2^r values. Furthermore symbols need not necessarily be restricted to 2^r values but can be defined for any positive integer.

Non-binary codes are concerned with the detection and correction of errors in symbols. The construction of non-binary codes, along with encoding and decoding techniques, follows directly from that of binary codes. The main difference arises in the need to determine the magnitude of errors and not just the error locations. In binary codes error magnitudes are 1 and it is only necessary to determine the position of errors. Once located error correction is achieved by simply inverting the

erroneous bits. With a non-binary code we first locate the position of the errors and then determine the magnitude of the errors.

Binary codes can be viewed as codes whose symbols have 2 values, 0 and 1, that is the code's symbols lie in $GF(2)$. A non-binary code has its symbols in the field $GF(q)$ where q is a prime number or any power of a prime number. A non-binary (n, k) linear code will have codewords of the form $\boldsymbol{c} = (c_{n-1}, c_{n-2}, \ldots, c_2, c_1, c_0)$ where the codeword components lie in $GF(q)$ and there exists at least one set of k codewords from which all the other codewords can be obtained by linear combinations of the k codewords. A non-binary (n, k) cyclic code can be constructed from a polynomial $g(x)$ of degree $n-k$, where $g(x)$ has its coefficients in $GF(q)$ and divides x^n-1. Note that for binary codes $x^n - 1 = x^n + 1$ and so $g(x)$ can be said to divide the latter if it is to generate a cyclic code.

A t-error-correcting nonbinary BCH code of blocklength $n = q^m - 1$ is a (n, k) cyclic code whose generator polynomial $g(x)$ has its coefficients in $GF(q)$ and roots

$$\beta, \beta^2, \ldots, \beta^{2t}$$

in $GF(q^m)$ an extension field of $GF(q)$. Recall that the generator polynomial of a t-error-correcting binary BCH code is given by the least common multiple LCM of the minimal polynomials $m_i(x)$, over $GF(2)$, of $2t$ consecutive field elements. To construct a non-binary BCH code minimal polynomials over $GF(q)$ are required and the generator polynomial of the code is given by

$$g(x) = \text{LCM}[m_1(x), m_2(x), \ldots, m_{2t}(x)]$$

where now $m_i(x)$ is the minimal polynomial over $GF(q)$ of β^i. If $q = 2$ then the minimal polynomials are binary and we obtain the binary BCH codes.

The most important class of non-binary BCH codes are the *Reed–Solomon codes*, which differ from other non-binary codes in that the base field and extension field are taken to be the same. Both the symbols and the generator polynomial roots lie in the field $GF(q)$ and define a Reed–Solomon code with blocklength $n = q - 1$. Here we consider Reed–Solomon codes where $q = 2^m$ and so symbols and roots lie in $GF(2^m)$. A t-error-correcting Reed–Solomon code is a cyclic code whose generator polynomial is the least-degree polynomial that has $\beta, \beta^2, \beta^3, \ldots, \beta^{2t}$ as roots, where β belongs to $GF(2^m)$. The minimal polynomial over $GF(2^m)$ of an element β in $GF(2^m)$ is the factor

$$m_\beta = x + \beta$$

as this is clearly the least-degree polynomial that has β as a root. The generator polynomial of a Reed–Solomon code is therefore

$$g(x) = (x + \beta)(x + \beta^2)(x + \beta^3) \cdots (x + \beta^{2t}). \tag{7.35}$$

Note that there is also no need to take the least common multiple of the factors as all the factors are distinct.

Consider a double-error-correcting Reed–Solomon code over $GF(2^4)$, taking $\beta = \alpha$ gives

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

and expanding this gives the generator polynomial

$$g(x) = x^4 + \alpha^{13}x^3 + \alpha^6x^2 + \alpha^3x + \alpha^{10}. \tag{7.36}$$

Note that the coefficients of $g(x)$ are no longer binary. The code's blocklength is $n = 2^4 - 1 = 15$, and as the degree of $g(x)$ is $r = 4$, then using $n - k = r$ gives $k = 11$ (recall that the degree of the generator polynomial of an (n, k) cyclic code is $n - k$). This is therefore the generator polynomial of a double-error-correcting (15, 11) Reed–Solomon code.

Example 7.7
Construct a single-error-correcting Reed–Solomon code with blocklength 7.

The code is constructed over $GF(2^3)$ as this gives a code with blocklength $n = 2^3 - 1 = 7$. Substituting $t = 1$ and $\beta = \alpha$ in eqn 7.35 gives

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

where α is a primitive element in $GF(2^3)$. The information length k is given by $k = n - r$, where $r = 2$ is the degree of $g(x)$ and $n = 7$, and so $k = 5$. This is therefore a single-error-correcting (7, 5) Reed–Solomon code. □

For encoding purposes the Reed–Solomon codes can be treated as cyclic codes or a generator matrix can be constructed from the generator polynomial and the codes can then be treated as linear codes. For example consider the (7, 5) Reed–Solomon code with generator polynomial

$$g(x) = x^2 + \alpha^4x + \alpha^3 \tag{7.37}$$

and let's construct the systematic codeword for the information word, say, $\boldsymbol{i} = (1\ 0\ \alpha\ \alpha^5\ \alpha^2)$ where α is an element of $GF(2^3)$. The information polynomial corresponding to $\boldsymbol{i}$ is

$$i(x) = x^4 + \alpha x^2 + \alpha^5x + \alpha^2$$

and multiplying this by $x^{n-k} = x^2$ gives

$$x^2i(x) = x^6 + \alpha x^4 + \alpha^5x^3 + \alpha^2x^2.$$

Recall that to construct systematic codewords we require the remainder of $x^{n-k}i(x)$ divided by $g(x)$. When dividing two non-binary polynomials, care has to be taken to ensure that at each step the coefficients of the highest power of x are the same. The division is a bit more awkward than that of dividing two binary polynomials,

however the principle is the same. Dividing $x^2 i(x)$ by $g(x)$ we get

$$
\begin{array}{r}
x^4 + \alpha^4 x^3 + \alpha^3 x^2 + \alpha^5 x + \alpha^6 \\
x^2 + \alpha^4 x + \alpha^3 \;\overline{)\; x^6 + \alpha x^4 + \alpha^5 x^3 + \alpha^2 x^2} \\
\underline{x^6 + \alpha^4 x^5 + \alpha^3 x^4} \\
\alpha^4 x^5 + x^4 + \alpha^5 x^3 + \alpha^2 x^2 \\
\underline{\alpha^4 x^5 + \alpha x^4 + x^3} \\
\alpha^3 x^4 + \alpha^4 x^3 + \alpha^2 x^2 \\
\underline{\alpha^3 x^4 + x^3 + \alpha^6 x^2} \\
\alpha^5 x^3 + x^2 \\
\underline{\alpha^5 x^3 + \alpha^2 x^2 + \alpha x} \\
\alpha^6 x^2 + \alpha x \\
\underline{\alpha^6 x^2 + \alpha^3 x + \alpha^2} \\
x + \alpha^2
\end{array}
$$

and the remainder is therefore $r(x) = x + \alpha^2$. Adding $r(x)$ to $x^2 i(x)$ gives the codeword polynomial

$$c(x) = x^2 i(x) + r(x) = x^6 + \alpha x^4 + \alpha^5 x^3 + \alpha^2 x^2 + x + \alpha^2$$

which gives the codeword $\boldsymbol{c} = (1\ 0\ \alpha\ \alpha^5\ \alpha^2\ 1\ \alpha^2)$.

Example 7.8

Construct the (15, 13) single-error-correcting Reed–Solomon code and determine the systematic codeword corresponding to $\boldsymbol{i} = (0\ 0\ \alpha\ 0\ 0\ 1\ \alpha^7\ \alpha^2\ 0\ 0\ 1\ \alpha\ \alpha^2)$ where α is a primitive element of $GF(2^4)$.

The generator polynomial is

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

Note that the degree of $g(x)$ is 2, which is consistent with the code's (n, k) parameters, $n - k = 15 - 13 = 2$. The information polynomial corresponding to $\boldsymbol{i}$ is

$$i(x) = \alpha x^{10} + x^7 + \alpha^7 x^6 + \alpha^2 x^5 + x^2 + \alpha x + \alpha^2$$

and dividing $x^2 i(x)$ by $g(x)$ gives the quotient and remainder

$$
\begin{aligned}
q(x) &= \alpha x^{10} + \alpha^6 x^9 + \alpha^{13} x^8 + \alpha^4 x^7 + \alpha^4 x^6 + \alpha^8 x^5 + \alpha^5 x^4 \\
&\quad + \alpha^{14} x^3 + \alpha^{10} x^2 + \alpha^{10} x + \alpha^3 \\
r(x) &= x\alpha^3 + \alpha^6
\end{aligned}
$$

respectively. The codeword polynomial is therefore

$$x^2 i(x) + r(x) = \alpha x^{12} + x^9 + \alpha^7 x^8 + \alpha^2 x^7 + x^4 + \alpha x^3 + \alpha^2 x^2 + \alpha^3 x + \alpha^6$$

which gives the codeword

$$\boldsymbol{c} = (0\ 0\ \alpha\ 0\ 0\ 1\ \alpha^7\ \alpha^2\ 0\ 0\ 1\ \alpha\ \alpha^2\ \alpha^3\ \alpha^6).$$

□

The generator polynomial $g(x)$ of a t-error-correcting Reed–Solomon code has $2t$ linear factors, one for each root $\beta, \beta^2, \ldots, \beta^{2t}$ and the degree of $g(x)$ is therefore $n - k = 2t$. Hence the number of parity-check symbols is $2t$ and as such a t-error correcting Reed–Solomon code with blocklength n can be referred to as a $(n, n - 2t)$ Reed–Solomon code. Recall that a t-error-correcting code requires a minimum distance of

$$d_{\min} = 2t + 1$$

and so a t-error-correcting Reed–Solomon code has

$$d_{\min} = n - k + 1$$

and therefore the Reed–Solomon codes are maximum-distance codes. Note also that the designed distance, d_0, and minimum distance, $d_{\min}$, of a Reed–Solomon code are the same.

The number of codewords in a non-binary code can be surprisingly large. An (n, k) binary code has 2^k codewords as there are 2^k distinct information words. For example the (7, 4) binary code has 16 codewords. In a t-error-correcting $(n, n - 2t)$ Reed–Solomon code over $GF(q)$ each information symbol has q distinct values and there are therefore q^{n-2t} codewords. For example each symbol of the single-error-correcting (7, 5) Reed–Solomon code has 8 distinct values and the code has 32 768 codewords. Clearly decoding algorithms that avoid the use of look-up tables are necessary with codes with such large numbers of codewords.

Decoding Reed–Solomon codes is achieved by first determining the error positions and then the error magnitudes. The methods used for locating errors in binary codes can also be used in Reed–Solomon codes, the only additional theory required is for finding error magnitudes. In a binary code an error pattern of μ errors can be represented by the error polynomial

$$e(x) = x^{p_1} + x^{p_2} + \cdots + x^{p_\mu}$$

where $p_1, p_2, \ldots, p_\mu$ are the error positions. Taking error magnitudes into account, the error polynomial becomes

$$e(x) = y_{p_1} x^{p_1} + y_{p_2} x^{p_2} + \cdots + y_{p_\mu} x^{p_\mu} \tag{7.38}$$

where y_{p_i} is the error magnitude at the position p_i. The decoder input is $v(x) = c(x) + e(x)$ where $c(x)$ is the codeword polynomial incurring the errors. For a t-error correcting code the error syndromes calculated by the decoder are

$$S_l = v(\alpha^l) = c(\alpha^l) + e(\alpha^l) = e(\alpha^l)$$

so giving

$$
\begin{aligned}
S_1 &= y_{p_1}\alpha^{p_1} + y_{p_2}\alpha^{p_2} + \cdots + y_{p_\mu}\alpha^{p_\mu} \\
S_2 &= y_{p_1}\alpha^{2p_1} + y_{p_2}\alpha^{2p_2} + \cdots + y_{p_\mu}\alpha^{2p_\mu} \\
S_3 &= y_{p_1}\alpha^{3p_1} + y_{p_2}\alpha^{3p_2} + \cdots + y_{p_\mu}\alpha^{3p_\mu} \\
&\vdots \\
S_{2t} &= y_{p_1}\alpha^{2tp_1} + y_{p_2}\alpha^{2tp_2} + \cdots + y_{p_\mu}\alpha^{2tp_\mu}.
\end{aligned} \tag{7.39}
$$

Recall that for binary codes we defined the error-location number $X_i = \alpha^{p_i}$. Here we define an additional term $Y_i = v_{p_i}$ known as the *error magnitude* of the ith error-location number, and in doing so we can express the syndrome equations in the more convenient form

$$\begin{aligned} S_1 &= Y_1X_1 + Y_2X_2 + \cdots + Y_\mu X_\mu \\ S_2 &= Y_1X_1^2 + Y_2X_2^2 + \cdots + Y_\mu X_\mu^2 \\ S_3 &= Y_1X_1^3 + Y_2X_2^3 + \cdots + Y_\mu X_\mu^3 \\ &\vdots \\ S_{2t} &= Y_1X_1^{2t} + Y_2X_2^{2t} + \cdots + Y_\mu X_\mu^{2t} \end{aligned} \tag{7.40}$$

We have already seen that for a binary code $S_2 = S_1^2$, this though does not apply to non-binary codes. Taking S_1^2 gives

$$\begin{aligned} S_1^2 &= (Y_1X_1 + Y_2X_2 + \cdots + Y_\mu X_\mu)^2 \\ &= Y_1^2X_1^2 + Y_2^2X_2^2 + \cdots + Y_\mu^2X_\mu^2 \neq S_2 \end{aligned}$$

Likewise we can show that $S_4 \neq S_2^2$ and clearly for a non-binary code

$$S_{2i} \neq S_i^2$$

and so the $2t$ error syndromes need to be individually evaluated.

Equations 7.40 consist of $2t$ equations with μ unknown error magnitudes, along with known error syndromes and error-location numbers, which can be solved to give the error magnitudes. Before addressing eqns 7.40 for any value of t we first consider a single-error-correcting code. The decoder of a single-error correcting code determines two syndromes S_1 and S_2. From eqn 7.40 setting $t = 1$ and $\mu = 1$ (as the maximum number of correctable errors is 1) gives

$$\begin{aligned} S_1 &= Y_1X_1 \\ S_2 &= Y_1X_1^2 \end{aligned} \tag{7.41}$$

and dividing S_2 by S_1 gives

$$\frac{S_2}{S_1} = \frac{Y_1X_1^2}{Y_1X_1} = X_1.$$

Substituting $X_1 = S_2/S_1$ into the first expression in eqns 7.41 gives $S_1 = Y_1(S_2/S_1)$ and so $Y_1 = S_1^2/S_2$. Therefore the error-location number X_1 and error magnitude Y_1 of a single-error correcting Reed–Solomon code are given by

$$\begin{aligned} X_1 &= S_2/S_1 \\ Y_1 &= S_1^2/S_2 \end{aligned} \tag{7.42}$$

Note that if we let $S_2 = S_1^2$, then eqns 7.42 give

$$\begin{aligned} X_1 &= S_1^2/S_1 = S_1 \\ Y_1 &= S_1^2/S_1^2 = 1 \end{aligned}$$

which are the correct error-location number and error magnitude for a single-error-correcting binary code.

Example 7.9

Consider the (7, 5) single-error-correcting Reed–Solomon code. Given that $\mathbf{r} = (0\ 1\ \alpha^5\ \alpha^2\ 1\ \alpha^6\ \alpha^3)$, where α is an element of $GF(2^3)$, corresponds to a codeword $\mathbf{c}$ with a single error, determine the position and magnitude of the error and the codeword $\mathbf{c}$.

The polynomial corresponding to $\mathbf{c}$ is

$$v(x) = x^5 + \alpha^5x^4 + \alpha^2x^3 + x^2 + \alpha^6x + \alpha^3$$

and in $GF(2^3)$ the error syndromes are

$$\begin{aligned} S_1 &= v(\alpha) = \alpha \\ S_2 &= v(\alpha^2) = \alpha^3. \end{aligned}$$

Using eqns 7.42 gives

$$\begin{aligned} X_1 &= S_2/S_1 = \alpha^3/\alpha = \alpha^2 \\ Y_1 &= S_1^2/S_2 = \alpha^2/\alpha^3 = \alpha^6 \end{aligned}$$

giving an error location of x^2 and error magnitude α^6. The error incurred by $\mathbf{c}$ is therefore α^6x^2 and so the codeword polynomial is

$$\begin{aligned} c(x) &= v(x) + \alpha^6x^2 \\ &= x^5 + \alpha^5x^4 + \alpha^2x^3 + (1 + \alpha^6)x^2 + \alpha^6x + \alpha^3 \\ &= x^5 + \alpha^5x^4 + \alpha^2x^3 + \alpha^2x^2 + \alpha^6x + \alpha^3 \end{aligned}$$

giving $\mathbf{c} = (0\ 1\ \alpha^5\ \alpha^2\ \alpha^2\ \alpha^6\ \alpha^3)$. □

A decoder for a t-error-correcting Reed–Solomon code determines the number of errors μ and the error-location numbers $X_1, X_2, \ldots, X_\mu$ using any technique that can be used for a binary BCH code. Once the error-location numbers have been found, the μ error magnitudes can be obtained by solving the first μ equations in eqns 7.40 for $Y_1, Y_2, \ldots, Y_\mu$. Note that the error syndromes, given by eqns 7.40, are nonlinear functions of the error-location numbers, but linear functions of the error magnitudes. Hence the error magnitudes can be obtained from the syndrome equations by using a standard matrix inversion method. Defining the column vectors $\boldsymbol{S}$ and $\boldsymbol{Y}$ as

$$\boldsymbol{S} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_\mu \end{bmatrix} \tag{7.43}$$

$$\boldsymbol{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_\mu \end{bmatrix} \tag{7.44}$$

and the matrix X as

$$X = \begin{bmatrix} X_1 & X_2 & X_3 & \cdots & X_\mu \\ X_1^2 & X_2^2 & X_3^2 & \cdots & X_\mu^2 \\ X_1^3 & X_2^3 & X_3^3 & \cdots & X_\mu^3 \\ & & & & \\ X_1^\mu & X_2^\mu & X_3^\mu & \cdots & X_\mu^\mu \end{bmatrix} \tag{7.45}$$

then eqns 7.40 can be written as

$$S = XY.$$

Note that the column vector S given by eqn 7.43 is not the same as S (eqn 7.32) defined for use in the Peterson–Gorenstein–Zierler decoder, for convenience the same notation is used, this should not cause confusion. The matrix X and column vector S are known terms, and so Y can be found by inverting $S = XY$ to give $Y = X^{-1}S$. Therefore the error magnitudes are given by

$$Y = X^{-1}S. \tag{7.46}$$

Note that X cannot be singular because μ nonzero and distinct errors are already known to exist.

Finding error magnitudes is simpler than we may have at first expected. The nonlinear problem faced when determining the error locations does not arise. Instead the error magnitudes are linearly related to the error syndromes and the known error-location numbers. However, as before, we face the computationally inefficient process of matrix inversion. Later we shall see how this matrix inversion can be circumvented (see Section 7.9). Decoding Reed–Solomon codes can be summarized as follows:

1. Find the number of errors μ and error-location numbers $X_1, X_2, \ldots, X_\mu$ by using any technique suitable to binary BCH codes.
2. From the error-location numbers construct the matrix X and determine its inverse X^{-1}.
3. The error magnitudes $Y_1, Y_2, \ldots, Y_\mu$ are then given by $Y = X^{-1}S$, where S is the column vector constructed from the error syndromes $S_1, S_2, S_3, \ldots, S_\mu$.

The example that follows considers decoding a (15, 9) triple-error-correcting Reed–Solomon code. The approach used is based on the Peterson–Gorenstein–Zierler decoder with two matrix inversions, one for the error-location numbers and the other for the error magnitudes. Consider a codeword polynomial $c(x)$, belonging to the triple-error-correcting Reed–Solomon (15, 9) code, that has incurred 3 errors so giving

$$v(x) = \alpha^3 x^{12} + x^8 + \alpha^{10} x^7 + \alpha^2 x^5 + \alpha^8 x^4 + \alpha^{14} x^3 + \alpha^6.$$

Over $GF(2^4)$ the error syndromes are

$$S_1 = v(\alpha) = \alpha^{13} + \alpha^8 + \alpha^{17} + \alpha^7 + \alpha^{12} + \alpha^{17} + \alpha^6 = \alpha^8$$
$$S_2 = v(\alpha^2) = \alpha^{27} + \alpha^{16} + \alpha^{24} + \alpha^{12} + \alpha^{16} + \alpha^{20} + \alpha^6 = 0$$
$$S_3 = v(\alpha^3) = \alpha^{39} + \alpha^{24} + \alpha^{31} + \alpha^{17} + \alpha^{20} + \alpha^{23} + \alpha^6 = \alpha^{14}$$
$$S_4 = v(\alpha^4) = \alpha^{51} + \alpha^{32} + \alpha^{38} + \alpha^{22} + \alpha^{24} + \alpha^{26} + \alpha^6 = \alpha^{11}$$
$$S_5 = v(\alpha^5) = \alpha^{63} + \alpha^{40} + \alpha^{45} + \alpha^{27} + \alpha^{28} + \alpha^{29} + \alpha^6 = \alpha^{14}$$
$$S_6 = v(\alpha^6) = \alpha^{75} + \alpha^{48} + \alpha^{52} + \alpha^{32} + \alpha^{32} + \alpha^{32} + \alpha^6 = \alpha^9.$$

The decoder first assumes that the maximum number of correctable errors, 3, have occurred and constructs the matrix

$$M = \begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} = \begin{bmatrix} \alpha^8 & 0 & \alpha^{14} \\ 0 & \alpha^{14} & \alpha^{11} \\ \alpha^{14} & \alpha^{11} & \alpha^{14} \end{bmatrix}$$

Evaluating the determinate of M gives

$$\det(M) = \alpha^8 \begin{vmatrix} \alpha^{14} & \alpha^{11} \\ \alpha^{11} & \alpha^{14} \end{vmatrix} + 0 \begin{vmatrix} 0 & \alpha^{11} \\ \alpha^{14} & \alpha^{14} \end{vmatrix} + \alpha^{14} \begin{vmatrix} 0 & \alpha^{14} \\ \alpha^{14} & \alpha^{11} \end{vmatrix}$$
$$= \alpha^8(\alpha^{13} + \alpha^7) + \alpha^{14}(\alpha^{14}\alpha^{14}) = \alpha^{11} + \alpha^{12} = 1$$

Hence $\det(M) \neq 0$ and so the decoder assumes that 3 errors have occurred (which we know is correct). The inverse of M is

$$M^{-1} = \frac{\text{adj}(M)}{\det(M)} = \begin{bmatrix} \alpha^5 & \alpha^{10} & \alpha^{13} \\ \alpha^{10} & \alpha^7 & \alpha^2 \\ \alpha^{13} & \alpha^2 & \alpha^5 \end{bmatrix}.$$

The coefficients of the error-location polynomial are given by

$$\begin{bmatrix} \sigma_3 \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha^5 & \alpha^{10} & \alpha^{13} \\ \alpha^{10} & \alpha^7 & \alpha^2 \\ \alpha^{13} & \alpha^2 & \alpha^5 \end{bmatrix} \begin{bmatrix} \alpha^{11} \\ \alpha^{14} \\ \alpha^9 \end{bmatrix} = \begin{bmatrix} \alpha^{16} + \alpha^{24} + \alpha^{22} \\ \alpha^{21} + \alpha^{21} + \alpha^{11} \\ \alpha^{24} + \alpha^{16} + \alpha^{14} \end{bmatrix} = \begin{bmatrix} \alpha^4 \\ \alpha^{11} \\ 1 \end{bmatrix}$$

and so $\sigma_1 = 1$, $\sigma_2 = \alpha^{11}$ and $\sigma_3 = \alpha^4$ giving the error-location polynomial

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \sigma_3 x^3 = 1 + x + \alpha^{11} x^2 + \alpha^4 x^3.$$

Searching $GF(2^4)$ for the roots of $\sigma(x)$ shows that $x = \alpha^3, \alpha^9$, and α^{14} are roots, and taking the reciprocals of the roots gives the error-location numbers $X_1 = \alpha^{12}$,

$X_2 = \alpha^8$, and $X_3 = \alpha$ respectively. To find the error magnitudes we construct

$$\boldsymbol{X} = \begin{bmatrix} X_1 & X_2 & X_3 \\ X_1^2 & X_2^2 & X_3^2 \\ X_1^3 & X_2^3 & X_3^3 \end{bmatrix} = \begin{bmatrix} \alpha^{12} & \alpha^6 & \alpha \\ \alpha^9 & \alpha^{12} & \alpha^2 \\ \alpha^6 & \alpha^3 & \alpha^3 \end{bmatrix}$$

The determinant of $\boldsymbol{X}$ is $\det(\boldsymbol{X}) = \alpha^2$ and its inverse is

$$\boldsymbol{X}^{-1} = \begin{bmatrix} \alpha^8 & \alpha^{12} & \alpha \\ \alpha^7 & \alpha^7 & \alpha^9 \\ \alpha^8 & \alpha^9 & \alpha^5 \end{bmatrix}$$

Using $\boldsymbol{Y} = \boldsymbol{X}^{-1}\boldsymbol{S}$ the error magnitudes Y_1, Y_2, and Y_3 are given by

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} \alpha^8 & \alpha^{12} & \alpha \\ \alpha^7 & \alpha^7 & \alpha^9 \\ \alpha^8 & \alpha^9 & \alpha^5 \end{bmatrix} \begin{bmatrix} \alpha^6 \\ 0 \\ \alpha^{14} \end{bmatrix} = \begin{bmatrix} \alpha^{14} + \alpha^{15} \\ \alpha^{13} + \alpha^{23} \\ \alpha^{14} + \alpha^{19} \end{bmatrix} = \begin{bmatrix} \alpha^3 \\ \alpha^3 \\ \alpha^9 \end{bmatrix}$$

and so $Y_1 = \alpha^3$, $Y_2 = \alpha^3$, and $Y_3 = \alpha^9$. The error-location numbers $X_1 = \alpha^{12}$, $X_2 = \alpha^6$, and $X_3 = \alpha$ correspond to errors in positions x^{12}, x^6, and x respectively, the error pattern is therefore

$$e(x) = \alpha^3 x^{12} + \alpha^3 x^6 + \alpha^9 x$$

and adding this to $v(x)$ gives the codeword polynomial

$$c(x) = x^8 + \alpha^{10}x^7 + \alpha^3 x^6 + \alpha^2 x^5 + \alpha^8 x^4 + \alpha^{14}x^3 + \alpha^9 x + \alpha^6.$$

Example 7.10

A codeword $c(x)$ belonging to the triple-error-correcting (15, 9) Reed–Solomon code incurs errors so giving

$$v(x) = x^{10} + \alpha^3 x^8 + \alpha^{11}x^7 + \alpha^8 x^6 + \alpha^6 x^5 + \alpha^4 x^4 + \alpha^5 x^2 + \alpha^9 x + \alpha^6$$

determine $c(x)$.

The error syndromes corresponding to $v(x)$ are

$$S_1 = \alpha^4, S_2 = 1, S_3 = \alpha^{10}, S_4 = \alpha^7, S_5 = 0, S_6 = \alpha^{14}$$

over $GF(2^4)$. Taking $\mu = 3$ and substituting the error syndromes into eqn 7.31 gives

$$\boldsymbol{M} = \begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} = \begin{bmatrix} \alpha^4 & 1 & \alpha^{10} \\ 1 & \alpha^{10} & \alpha^7 \\ \alpha^{10} & \alpha^7 & 0 \end{bmatrix}$$

the inverse of which is found to be

$$\boldsymbol{M}^{-1} = \begin{bmatrix} 1 & \alpha^3 & \alpha^{14} \\ \alpha^3 & \alpha^6 & 1 \\ \alpha^{14} & 1 & \alpha^4 \end{bmatrix}.$$

Furthermore

$$\boldsymbol{S} = \begin{bmatrix} S_4 \\ S_5 \\ S_6 \end{bmatrix} = \begin{bmatrix} \alpha^7 \\ 0 \\ \alpha^{14} \end{bmatrix}$$

and substituting $\boldsymbol{M}^{-1}$ and $\boldsymbol{S}$ into eqn 7.34 gives the coefficients of the error-location polynomial

$$\begin{bmatrix} \sigma_3 \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \boldsymbol{M}^{-1}\boldsymbol{S} = \begin{bmatrix} 1 & \alpha^3 & \alpha^{14} \\ \alpha^3 & \alpha^6 & 1 \\ \alpha^{14} & 1 & \alpha^4 \end{bmatrix} \begin{bmatrix} \alpha^7 \\ 0 \\ \alpha^{14} \end{bmatrix} = \begin{bmatrix} \alpha^7 \\ \alpha^{10} \\ \alpha^2 \end{bmatrix}$$

Therefore the error-location polynomial is

$$\sigma(x) = 1 + \alpha^2 x + \alpha^{10}x^2 + \alpha^5 x^3$$

the roots of which are α^5, α^8, α^{12} which correspond to error-location numbers $X_1 = \alpha^{10}$, $X_2 = \alpha^7$, and $x_3 = \alpha^3$ respectively. To determine the error magnitudes we construct

$$\boldsymbol{X} = \begin{bmatrix} X_1 & X_2 & X_3 \\ X_1^2 & X_2^2 & X_3^2 \\ X_1^3 & X_2^3 & X_3^3 \end{bmatrix} = \begin{bmatrix} \alpha^{10} & \alpha^7 & \alpha^3 \\ \alpha^5 & \alpha^{14} & \alpha^6 \\ 1 & \alpha^5 & \alpha^9 \end{bmatrix}$$

and its inverse

$$\boldsymbol{X}^{-1} = \begin{bmatrix} \alpha^{12} & \alpha^6 & \alpha^2 \\ \alpha^{11} & \alpha^{10} & \alpha^{13} \\ \alpha^{13} & \alpha^2 & \alpha^{11} \end{bmatrix}$$

along with

$$\boldsymbol{S} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} \alpha^4 \\ 1 \\ \alpha^{10} \end{bmatrix}$$

Using eqn 7.46 the error magnitudes are given by

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} \alpha^{12} & \alpha^6 & \alpha^2 \\ \alpha^{11} & \alpha^{10} & \alpha^{13} \\ \alpha^{13} & \alpha^2 & \alpha^{11} \end{bmatrix} \begin{bmatrix} \alpha^4 \\ 1 \\ \alpha^{10} \end{bmatrix} = \begin{bmatrix} 1 \\ \alpha^4 \\ \alpha^5 \end{bmatrix}$$

and so $Y_1 = 1$, $Y^2 = \alpha^4$, and $Y_3 = \alpha^5$. The error-location numbers $X_1 = \alpha^{10}$, $X_2 = \alpha^7$, and $X_3 = \alpha^3$ correspond to errors in positions x^{10}, x^7, and x^3 respectively, the error pattern is therefore

$$e(x) = x^{10} + \alpha^4 x^7 + \alpha^5 x^3$$

and adding this to $r(x)$ gives

$$c(x) = \alpha^3x^5 + \alpha^{-1}x^7 + \alpha^8x^5 + \alpha^6x^3 + \alpha^4x^4 + \alpha^6x^3 + \alpha^5x^2 + \alpha^9x + \alpha^8$$

as the required codeword polynomial. □

7.8 The Berlekamp algorithm

The Peterson–Gorenstein–Zierler decoder is fundamental to decoding BCH codes, it overcomes the problem of solving the nonlinear syndrome equations by the use of an error-location polynomial whose coefficients can be obtained by standard linear matrix-inversion methods. This though is the weak link in the decoder, for matrix inversion requires excessive computation, especially for large matrices. In particular for non-binary codes for which a second matrix inversion is required to obtain the error magnitudes. As such the Peterson–Gorenstein–Zierler decoder is quite inefficient and whilst it is of prime importance in illustrating the principles of decoding BCH codes it is, nevertheless, of limited practical use.

The Berlekamp algorithm is a fast and efficient algorithm for decoding BCH codes. Like the Peterson–Gorenstein–Zierler decoder it uses the error-location polynomial, but it avoids the need for matrix inversion when determining the polynomial coefficients. The algorithm is generally considered to be significantly more complex than the Peterson–Gorenstein–Zierler decoder. However its complexity lies mainly in the proof of the algorithm, which we omit.

The algorithm uses an iterative technique to find an error-location polynomial $\sigma(x)$ whose coefficients satisfy Newton's identities, as given by eqns 7.23 and 7.24. The algorithm starts by finding a polynomial $\sigma^{(1)}(x)$, whose coefficients satisfy the first of Newton's identities. A suitable set of initial conditions is required to achieve this, otherwise the algorithm may fail to carry out the required number of iterations. The polynomial $\sigma^{(1)}(x)$ must not only satisfy the first identity, but it must be the polynomial of least degree that meets the requirement. Note that the superscript 1 in $\sigma^{(1)}(x)$ is enclosed in parenthesis to avoid any possible ambiguity with powers of $\sigma(x)$. Next a polynomial $\sigma^{(2)}(x)$ is found that satisfies the first and second identities, again the polynomial must be the polynomial of least degree that meets the requirement. To find $\sigma^{(2)}(x)$ we first check if $\sigma^{(1)}(x)$ meets the requirement, if it does then we let $\sigma^{(2)}(x) = \sigma^{(1)}(x)$. Otherwise $\sigma^{(1)}(x)$ is modified by adding a suitable correction term such that the resulting polynomial has coefficients that satisfy the first two equations of Newton's identities. The modification to $\sigma^{(1)}(x)$ must ensure that $\sigma^{(2)}(x)$ is the polynomial of least degree that meets the requirements. The process continues iteratively. Check to see if $\sigma^{(2)}(x)$ satisfies the first three identities, if it does then $\sigma^{(3)}(x) = \sigma^{(2)}(x)$, otherwise modify $\sigma^{(2)}(x)$ to obtain $\sigma^{(3)}(x)$. Then generate $\sigma^{(4)}(x)$, $\sigma^{(5)}(x)$, . . and so forth, until a polynomial $\sigma^{(2t)}(x)$, with coefficients satisfying all the Newton's identities, is obtained. The error-location polynomial is then

$$\sigma(x) = \sigma^{(2t)}(x)$$

and the error-location numbers are found in the usual manner of finding the inverse roots of $\sigma(x)$.

Recall that Newton's identities relate the error syndromes to the coefficients of the error-location polynomial. At each iteration in the algorithm the polynomial coefficients are used to estimate the error syndrome of following iteration. Let $\sigma^{(i)}(x)$ be the polynomial of least degree whose coefficients satisfy the first i Newton's identities, we can express $\sigma^{(i)}(x)$ as

$$\sigma^{(i)}(x) = 1 + \sigma_1^{(i)}(x) + \sigma_2^{(i)}x^2 + \cdots + \sigma_{r_i}^{(i)}x^{r_i}$$

where r_i is the degree of $\sigma^{(i)}(x)$. To test whether the coefficients of $\sigma^{(i)}(x)$ satisfy the $(i+1)$th identity we compute

$$\hat{S}_{i+1} = \sigma_1^{(i)}S_i + \sigma_2^{(i)}S_{i-1} + \cdots + \sigma_{r_i}^{(i)}S_{i+1-r_i} \tag{7.47}$$

which can be thought of as the $(i+1)$th syndrome as estimated or predicted by $\sigma^{(i)}(x)$. Adding $\hat{S}_{i+1}$ to the error syndrome S_{i+1} gives

$$d_i = S_{i+1} + \hat{S}_{i+1} \tag{7.48}$$

where d_i is known as the *ith discrepancy*. Both S_{i+1} and $\hat{S}_{i+1}$ are elements in the same field $GF(2^m)$ and the discrepancy is therefore a field element that gives a measure of the difference between the two error syndromes (as its name obviously implies). If $d_i = 0$ then the coefficients of $\sigma^{(i)}(x)$ satisfy the first $(i+1)$ identities and $\sigma^{(i)}(x)$ is taken as the next polynomial $\sigma^{(i+1)}(x)$, and therefore

$$\sigma^{(i+1)}(x) = \sigma^{(i)}(x) \tag{7.49}$$

The degree r_{i+1} of $\sigma^{(i+1)}(x)$ is the same as that of $\sigma^{(i)}(x)$, and so

$$r_{i+1} = r_i.$$

If $d_i \neq 0$ then the coefficients of $\sigma^{(i)}(x)$ fail to satisfy the $(i+1)$th identity and $\sigma^{(i)}(x)$ has to be modified by adding a suitable correction term. The correction term is a polynomial that depends upon one of the previous polynomials $\sigma^{(k)}(x)$, such that:

(1) the discrepancy $d_k \neq 0$;

(2) n_k has the largest value

where $n_k = k - r_k$ and r_k is the degree of $\sigma^{(k)}(x)$. The polynomial required is then given by

$$\sigma^{(i+1)}(x) = \sigma^{(i)}(x) + \left(\frac{x^l d_i}{x^k d_k}\right)\sigma^k(x) \tag{7.50}$$

and is the polynomial of least degree whose coefficients satisfy Newton's identities up to the $(i+1)$th. The correction term in eqn 7.50 has degree $i + r_k - k = i - n_k$ and so r_{i+1}, the degree of $\sigma^{(i+1)}(x)$, is r_i or $i - n_k$ depending upon which has the largest