Coding and cryptography

Additional material on Reed-Solomon codes

We clarify some of the material in Sections 6.1 through 6.4 of the textbook. In particular, we also discuss why, if the algorithms presented there result in a word, this is a codeword.

1 Linear codes over Galois fields

Let $F = GF(2^r)$ for $r \geq 2$ (or even $r = 1$, when we have K). On F^n we also have the Hamming distance $d(v, w)$ which counts in how many positions v and w differ, and the weight $\mathrm{wt}(w)$ of w, which is the number of non-zero positions of w, i.e., $\mathrm{wt}(w) = d(0, w)$. The distance still has the following properties:

1. $d(v, w) \geq 0$ for all v and w, and $d(v, w) = 0$ if and only if $v = w$;
2. $d(v, w) = d(w, v)$ for all v and w;
3. $d(v, w) \leq d(v, u) + d(u, w)$ for all u, v and w.

We can define the distance $d(C) = \min\{d(v, v') \text{ with } v \neq v' \text{ in } C\}$ for any code C in F^n. If $t = \lfloor \frac{d(C)-1}{2} \rfloor$ then C is t-error correcting in the sense that if v is sent and an error e of weight at most t occurs, so $w = v + e$ is received, then v is the unique nearest neighbour in C to w for the distance on F^n, and for some error e of weight $t + 1$ and some v, v is not the unique nearest neighbour to $w = v + e$ (either because there is no unique nearest neighbour in C to w or because there is a unique nearest neighbour but it is not v).

A linear code C of length n over a field F is a linear subspace (over F) of F^n. For a linear code C we have $d(C) = \min\{\mathrm{wt}(v) \text{ with } v \neq 0 \text{ in } C\}$.

A linear code C can always be defined by a check matrix H, which, if C has length n and dimension k, is an $n \times (n-k)$ matrix (with entries in F) with linearly independent columns, and we again have that $d(C)$ equals the smallest number of rows in H that are linearly dependent over F. The proof of the Singleton bound goes through without change, so for a linear (n, k, d)-code in F^n we always have $d - 1 \leq n - k$. If this bound is attained we call the code MDS as before.

Example 1.1. We take $r = 2$, and $F = GF(4) = \{0, 1, \alpha, \alpha^2\}$ with $\alpha^2 = 1 + \alpha$ and $\alpha^3 = 1$. We use $m = 0$ and $\beta = \alpha$. If we take $\delta = 2$ then $g(x) = \alpha + x$, and

$$C = \{a(x)(\alpha + x) \text{ with } a(x) \text{ in } F[x] \text{ with } \deg(a(x)) < 2\},$$

which has dimension 2 and distance 2. If we take $\delta = 3$ then $g(x) = (\alpha+x)(\alpha^2+x) = 1+x+x^2$, and we obtain the code

$$C' = \{a(x)(1 + x + x^2) \text{ with } a(x) \text{ in } F[x] \text{ with } \deg(a(x)) < 1\},$$

which is contained in C, and which has dimension 1 (it is the triple repetition code over F) and distance 3.

Definition 1.2. To a code $C \subseteq F^n$ with $F = GF(2^r)$ we can associate two codes over K:

(a) the subfield code $C_K = C \cap K^n$;

(b) the associated binary code $\widehat{C}$ in K^{rn} that we obtain by replacing every position in F of a word in C with the corresponding element in K^r in a fixed table of F (the table depends on the polynomial defining F and the choice of a primitive element of F).

Note that $d(C_K) \geq d(C)$ simply because $C_K \subseteq C$, and $d(\widehat{C}) \geq d(C)$ because different elements in F lead to different elements of K^r (but we also have $rd(C) \geq d(\widehat{C})$).

Example 1.3. For the code C in Example 1.1 we have $C_K = \{000, 111\}$: if $w(x)$ in $K[x]$ has root α, which is the condition to be in C, then it also has root α^2 because $w(x)^2 = w(x^2)$. Therefore $C_K = C'_K$ and the latter is clearly $\{000, 111\}$. Note that $d(C_K) = 3 > 2 = d(C)$.

For the associated binary codes, the sizes are the same of the original codes, so $\widehat{C}$ has $4^2 = 16$ elements and $\widehat{C'}$ has $4^1 = 4$ elements. For example, $1\alpha 0$ in C becomes 100100 in $\widehat{C}$ because in the table we have $1 \leftrightarrow 10$, $\alpha \leftrightarrow 01$ and $0 \leftrightarrow 00$.

2 Reed-Solomon codes

We fix a field $F = GF(2^r) = \{0, 1, \alpha, \alpha^2\alpha^3, \ldots, \alpha^{2^r-2}\}$ with primitive element α, where $r \geq 2$. Let n be a divisor of $2^r - 1$, and β non-zero in F with $1, \beta, \beta^2, \beta^3, \ldots, \beta^{n-1}$ all distinct as well as $\beta^n = 1$. Such a β always exist for a given n dividing $2^r - 1$ because we can take $\alpha^{(2^r-1)/n}$; in fact, the number of possible β equals $\varphi(n)$, with φ Euler's totient function.

Lemma 2.1. In $F[x]$ we have $1 + x^n = (1 + x)(\beta + x)(\beta^2 + x) \ldots (\beta^{n-1} + x)$.

Proof. Every β^i for $i = 0, 1, 2, \ldots, n - 1$ is a root of $1 + x^n$ because $\beta^n = 1$, so every $\beta^i + x$ for those i is a factor of $1 + x^n$. But the β^i are distinct, so those factors are distinct. Because there are n of those, their product equals $1 + x^n$ up to multiplication by a non-zero constant. Comparing the leading coefficients, we see this constant is 1. □

Remark 2.2. That $\beta^n = 1$ and $1, \beta, \beta^2, \beta^3, \ldots, \beta^{n-1}$ are distinct already implies that n is a divisor of $2^r - 1$, as follows. We always have $\beta^{2^r-1} = 1$, so because of Bézout's formula for the gcd, we also have $\beta^{\gcd(n,2^r-1)} = 1$. But $\gcd(n, 2^r - 1) < n$ is not possible because we assumed that $1, \beta, \beta^2, \beta^3, \ldots, \beta^{n-1}$ are distinct. Therefore $\gcd(2^r - 1, n) \geq n$, hence $\gcd(2^r - 1, n) = n$, and n divides $2^r - 1$.

Remark 2.3. The book first assumes that $n = 2^r - 1$ and later on says that everything would have worked for n dividing $2^r - 1$. Here we start directly with n dividing $2^r - 1$.

Definition 2.4. Given F and β as above, we define the Reed-Solomon code of length n, for m in $\mathbb{Z}$, and $\delta = 2, 3, \ldots, n$, to be the cyclic linear code over F with generator polynomial

$$g(x) = (\beta^{m+1} + x)(\beta^{m+2} + x) \ldots (\beta^{m+\delta-1} + x)\,.$$

As a shorthand we denote it $RS(n, \delta)$, even though it also depends on β and m.

Remark 2.5. We could include the cases $\delta = 1$ and $\delta = n + 1$, but the former leads to F^n as the code, and the latter to $\{0\}$, so we ignore those.

Remark 2.6. By Lemma 2.1, $g(x)$ divides $1 + x^n$ in $F[x]$. Just as for K, the cyclic linear codes over F of length n can be interpreted as the linear subspaces (over F) of $F[x]$ modulo $1 + x^n$ that are closed under multiplication by x. If this is not the zero subspace, then there is a unique monic[1] polynomial $g(x)$ of degree less than n in the code, which generates the code, and it has to divide $1 + x^n$ in $F[x]$. Conversely, if $g(x)$ is monic and divides $1 + x^n$ in $F[x]$, then it is the generator polynomial of a cyclic linear code: if $\deg(g(x)) = n - k$ then the code is $\{a(x)g(x) \text{ with } \deg(a(x)) < k\}$, which has dimension k. All those statements are proved as over K (their proofs depend only on division with remainder), the only modification that is needed is to make the generator polynomial unique by demanding it to be monic.

[1] This means that the leading coefficient is 1, which is automatic if the field is K.

Note that $\deg(g(x)) = \delta - 1$, so

$$RS(n,\delta) = \{a(x)g(x) \text{ with } a(x) \text{ in } F[x] \text{ and } \deg(a(x)) < n - \delta + 1\},$$

so that $RS(n,\delta)$ has dimension $n-\delta+1$. Also, $w(x)$ in $F[x]$ of degree less than n is in $RS(n,\delta)$ if and only if $s_j = 0$ for j in J, where we let $s_j = w(\beta^j)$ and $J = \{m+1, m+2, \ldots, m+\delta-1\}$. As before we call those s_j the syndromes of $w(x)$.

Proposition 2.7. Let all the notation be as above.

1. The distance of $RS(n,\delta)$ equals δ.
2. The dimension of $RS(n,\delta)$ equals $n-\delta+1$; in particular, this code is MDS.

Proof. We claim that a check matrix of this code is the $n \times (\delta-1)$ matrix

$$H = (\beta^{ij})_{\substack{i=0,\ldots,n-1 \\ j=m+1,\ldots,m+\delta-1}} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ \beta^{m+1} & \beta^{m+2} & \beta^{m+3} & \ldots & \beta^{m+\delta-1} \\ \beta^{2(m+1)} & \beta^{2(m+2)} & \beta^{2(m+3)} & \ldots & \beta^{2(m+\delta-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \beta^{(n-1)(m+1)} & \beta^{(n-1)(m+2)} & \beta^{(n-1)(m+3)} & \ldots & \beta^{(n-1)(m+\delta-1)} \end{pmatrix}$$

as the dot product of a vector $w = w_0 w_1 \ldots w_{n-1}$ with the jth row of this matrix is equal to $w_0 + w_1\beta^{m+j} + w_2\beta^{2(m+j)} + \cdots + w_{n-1}\beta^{(n-1)(m+j)}$, which equals $w(\beta^{m+j})$ for the polynomial $w_0 + w_1 x + w_2 x^2 + \cdots + w_{n-1}x^{n-1}$. So w is in the code defined by H if and only if $w(\beta^j) = 0$ for j in J, which is the same as saying that $w(x)$ is divisible by $\beta^j + x$ for those j. As all β^j with j in J are distinct, this is equivalent to $w(x)$ being divisible by $g(x)$ as above. So the nullspace of H is the code. That the conditions imposed by the rows are independent follows from considering the dimension of the resulting code, which we saw already to be $n-\delta+1$. It also follows from the calculations for the distance below, without using polynomials at all.

Note that any δ rows of H, which lie in $F^{\delta-1}$ must be dependent over F. So we only have to show that any $\delta-1$ rows of H are linearly independent (which will also show that the $\delta-1$ columns of H are linearly independent). Let $i_1, \ldots, i_{\delta-1}$ be different elements in $\{0, 1, \ldots, n-1\}$. Then the rows indexed by those are the rows in

$$\begin{pmatrix} \beta^{i_1(m+1)} & \beta^{i_1(m+2)} & \beta^{i_1(m+3)} & \ldots & \beta^{i_1(m+\delta-1)} \\ \beta^{i_2(m+1)} & \beta^{i_2(m+2)} & \beta^{i_2(m+3)} & \ldots & \beta^{i_2(m+\delta-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \beta^{i_{\delta-1}(m+1)} & \beta^{i_{\delta-1}(m+2)} & \beta^{i_{\delta-1}(m+3)} & \ldots & \beta^{i_{\delta-1}(m+\delta-1)} \end{pmatrix}.$$

Taking out the factor $\beta^{i_j(m+1)}$ from the jth row here for $j = 1, 2, \ldots, \delta-1$, we see that the determinant of this matrix is the product of $\beta^{(m+1)(i_1+i_2+\cdots+i_{\delta-1})}$ and the determinant of

$$\begin{pmatrix} 1 & \beta^{i_1} & \beta^{2i_1} & \ldots & \beta^{(\delta-2)i_1} \\ 1 & \beta^{i_2} & \beta^{2i_2} & \ldots & \beta^{(\delta-2)i_2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \beta^{i_{\delta-1}} & \beta^{2i_{\delta-1}} & \ldots & \beta^{(\delta-2)i_{\delta-1}} \end{pmatrix}.$$

The power of β is non-zero because β is non-zero. The determinant of the last matrix is non-zero by Proposition 5.1 because $\beta^{i_1}, \ldots, \beta^{i_{\delta-1}}$ are distinct, as those elements are among the n distinct elements $1, \beta, \beta^2, \beta^3, \ldots, \beta^{n-1}$. So any $\delta-1$ rows of H are linearly independent over F.

We had already observed that some (in fact: any) δ rows of H were linearly dependent, so the distance of the code is δ.

As mentioned above, this argument with determinants also shows that the columns of H are linearly independent over F, so the dimension of the code is $k = n - (\delta - 1) = n - \delta + 1$. Therefore $\delta - 1 = n - k$, with δ the distance, and the code is MDS. □

Example 2.8. We take $r = 4$, so $F = GF(2^4)$, and we choose $F = \{0, 1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{14}\}$ with $\alpha^4 = 1 + \alpha$ (so F is obtained using the primitive irreducible polynomial $x^4 + x + 1$) and $\alpha^{15} = 1$. Then n has to divide 15.

1. Let us take $n = 3$, and $\beta = \alpha^5$. Then $1, \beta, \beta^2$ are distinct and $\beta^3 = 1$. If we take $m = -1$ and $\delta = 3$, then the generator polynomial is $g(x) = (1 + x)(\beta + x) = (1 + x)(\alpha^5 + x)$, and the code has dimension $n - \delta + 1 = 1$. A check matrix is (but one never needs it in practice, and in a larger example it would be enormous anyway)

$$H = \begin{pmatrix} 1 & 1 \\ 1 & \beta \\ 1 & \beta^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & \alpha^5 \\ 1 & \alpha^{10} \end{pmatrix} .$$

   For $w = w_0 w_1 w_2$ in F^3 we have $wH = w(1)w(\beta)$ in F^2, with $w(x) = w_0 + w_1 x + w_2 x^2$.

2. Now take $n = 5$, and $\beta = \alpha^3$. Then $1, \beta, \beta^2, \beta^3, \beta^4$ are distinct, and $\beta^5 = 1$. For $m = 0$ and $\delta = 4$, the generator polynomial is $g(x) = (\beta+x)(\beta^2+x)(\beta^3+x) = (\alpha^3+x)(\alpha^6+x)(\alpha^9+x)$, and the code has dimension $n - \delta + 1 = 2$. A check matrix is

$$H = \begin{pmatrix} 1 & 1 & 1 \\ \beta & \beta^2 & \beta^3 \\ \beta^2 & \beta^4 & \beta^6 \\ \beta^3 & \beta^6 & \beta^9 \\ \beta^4 & \beta^8 & \beta^{12} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ \alpha^3 & \alpha^6 & \alpha^9 \\ \alpha^6 & \alpha^{12} & \alpha^{18} \\ \alpha^9 & \alpha^{18} & \alpha^{27} \\ \alpha^{12} & \alpha^{24} & \alpha^{36} \end{pmatrix} .$$

   For $w = w_0 w_1 w_2 w_3 w_4$ in F^5 we in this case have $wH = w(\beta)w(\beta^2)w(\beta^3)$ in F^3, where $w(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$.

3 Decoding Reed-Solomon codes: first method

Keep the notation for F, n, β, m and δ as in Definition 2.4, so we have a Reed-Solomon code $RS(n, \delta)$ with generator polynomial $g(x) = (\beta^{m+1} + z)(\beta^{m+2} + z) \ldots (\beta^{m+\delta-1} + z)$, and we know that it can correct any $t = \lfloor \frac{\delta-1}{2} \rfloor$ errors. In this section we discuss a method for correcting any t errors if δ is odd (but see Remark 3.10 for the case that δ is even.)

Suppose a codeword $v(x)$ is received as $w(x)$. If the syndromes $s_j = w(\beta^j)$ of $w(x)$ are 0 for j in $J = \{m + 1, m + 2, \ldots, m + \delta - 1\}$, then $w(x)$ is in the code, and we accept it. If that is not the case, suppose $w(x) = v(x) + e(x)$ with

$$\begin{aligned} e(x) &= e_0 + e_1 x + \cdots + e_{n-1} x^{n-1} \\ &= \sum_{i=1}^{l} b_i x^{c_i} \end{aligned}$$

an error polynomial of weight l with $1 \leq l \leq t$, so that all b_i in F are non-zero, and all c_i in $\{0, 1, 2, \ldots, n - 1\}$ distinct. We shall determine $e(x)$ from the s_j with j in J.

For this, write $a_i = \beta^{c_i}$. Because of our assumptions on β and the c_i, all a_i are distinct, and in fact, we can determine c_i from a_i. We then define the *error locator polynomial* in $F[z]$ by

$$\begin{aligned}\sigma(z) &= (a_1 + z)(a_2 + z)\dots(a_l + z)\\ &= \sigma_0 + \sigma_1 z + \sigma_2 z^2 + \dots + \sigma_{l-1} z^{l-1} + z^l\end{aligned}$$

with the last line introducing notation for the coefficients of $\sigma(z)$. The c_i are the locations of the errors in $w(x)$, and each c_i is determined by $a_i = \beta^{c_i}$, which justifies the name.

Note that $s_j = w(\beta^j) = v(\beta^j) + e(\beta^j) = e(\beta^j) = \sum_{i=1}^{l} b_i(\beta^j)^{c_i} = \sum_{i=1}^{l} b_i a_i^j$ for j in J because $v(x)$ is in the code, and the definition of the a_i. We have $\sigma(a_i) = 0$ because of the factor $a_i + z$ in $\sigma(z)$. So for all $i = 1, 2, \dots, l$, and all integers j, we have

$$0 = \sigma(a_i) b_i a_i^j = \sigma_0 b_i a_i^j + \sigma_1 b_i a_i^{j+1} + \sigma_2 b_i a_i^{j+2} + \dots + \sigma_{l-1} b_i a_i^{j+l-1} + b_i a_i^{j+l}\,.$$

Summing this for $i = 1, 2, \dots, l$ and using that $s_k = \sum_{i=1}^{l} b_i a_i^k$ for all k in J gives

$$0 = \sigma_0 s_j + \sigma_1 s_{j+1} + \sigma_2 s_{j+2} + \dots + \sigma_{l-1} s_{j+l-1} + s_{j+l} \tag{3.1}$$

for all $j = m+1, m+2, \dots, m+\delta-l-1$ (we need $j, j+1, \dots, j+l$ to be in J). Because $l \le t$, and $2t \le \delta - 1$, we certainly have (3.1) for $j = m+1, m+2, \dots, m+l$. These identities are equivalent to one matrix equation

$$\begin{pmatrix} s_{m+1} & s_{m+2} & s_{m+3} & \dots & s_{m+l} \\ s_{m+2} & s_{m+3} & s_{m+4} & \dots & s_{m+l+1} \\ s_{m+3} & s_{m+4} & s_{m+5} & \dots & s_{m+l+2} \\ \vdots & \vdots & \vdots & & \vdots \\ s_{m+l} & s_{m+l+1} & s_{m+l+2} & \dots & s_{m+2l-1} \end{pmatrix} \begin{pmatrix} \sigma_0 \\ \sigma_1 \\ \vdots \\ \sigma_{l-1} \end{pmatrix} = \begin{pmatrix} s_{m+l+1} \\ s_{m+l+2} \\ \vdots \\ s_{m+2l} \end{pmatrix}.$$

Solving this for $\sigma_0, \sigma_1, \dots, \sigma_{l-1}$ would give us $\sigma(z)$. It is customary to abbreviate such systems to one (extended) matrix

$$(*) \qquad M_l' = \left(\begin{array}{ccccc|c} s_{m+1} & s_{m+2} & s_{m+3} & \dots & s_{m+l} & s_{m+l+1} \\ s_{m+2} & s_{m+3} & s_{m+4} & \dots & s_{m+l+1} & s_{m+l+2} \\ s_{m+3} & s_{m+4} & s_{m+5} & \dots & s_{m+l+2} & s_{m+l+3} \\ \vdots & \vdots & \vdots & & & \vdots \\ s_{m+l} & s_{m+l+1} & s_{m+l+2} & \dots & s_{m+2l-1} & s_{m+2l} \end{array}\right),$$

which suppresses $\sigma_0, \sigma_1, \dots, \sigma_{l-1}$. In order to write it down, we need to know l.

For this, we consider the matrix

$$M_l = \begin{pmatrix} s_{m+1} & s_{m+2} & s_{m+3} & \dots & s_{m+l} \\ s_{m+2} & s_{m+3} & s_{m+4} & \dots & s_{m+l+1} \\ s_{m+3} & s_{m+4} & s_{m+5} & \dots & s_{m+l+2} \\ \vdots & \vdots & \vdots & & \vdots \\ s_{m+l} & s_{m+l+1} & s_{m+l+2} & \dots & s_{m+2l-1} \end{pmatrix},$$

so that M_l' is obtained by extending M_l. This last matrix can be written as

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_l \\ \vdots & \vdots & & \vdots \\ a_1^{l-1} & a_2^{l-1} & \dots & a_l^{l-1} \end{pmatrix} \begin{pmatrix} b_1 a_1^{m+1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b_l a_l^{m+1} \end{pmatrix} \begin{pmatrix} 1 & a_1 & \dots & a_1^{l-1} \\ 1 & a_2 & \dots & a_2^{l-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_l & \dots & a_l^{l-1} \end{pmatrix}. \tag{3.2}$$

Because the a_i are distinct, from the properties of Vandermonde matrices (see Proposition 5.1), the two matrices on the outside (which are the transposed of each other) have non-zero determinants, and the middle does have non-zero determinant because all b_i and a_i are non-zero. This shows that $(*)$ has a unique solution, corresponding to the coefficients $\sigma_0, \ldots, \sigma_{l-1}$ of $\sigma(z)$. We can write down this system and determine $\sigma(z)$ if we know l, because all the s_j are known.

In order to finally find l, replace l with t, so we obtain the matrix

$$M_t = \begin{pmatrix} s_{m+1} & s_{m+2} & s_{m+3} & \cdots & s_{m+t} \\ s_{m+2} & s_{m+3} & s_{m+4} & \cdots & s_{m+t+1} \\ s_{m+3} & s_{m+4} & s_{m+5} & \cdots & s_{m+t+2} \\ \vdots & \vdots & \vdots & & \vdots \\ s_{m+t} & s_{m+t+1} & s_{m+t+2} & \cdots & s_{m+2t-1} \end{pmatrix},$$

which can similarly be written as

$$\begin{pmatrix} 1 & 1 & \ldots & 1 \\ a_1 & a_2 & \ldots & a_t \\ \vdots & \vdots & & \vdots \\ a_1^{t-1} & a_2^{t-1} & \ldots & a_t^{t-1} \end{pmatrix} \begin{pmatrix} b_1 a_1^{m+1} & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & b_t a_t^{m+1} \end{pmatrix} \begin{pmatrix} 1 & a_1 & \ldots & a_1^{t-1} \\ 1 & a_2 & \ldots & a_2^{t-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_t & \ldots & a_t^{t-1} \end{pmatrix}$$

if we use $b_{l+1}, \ldots, b_t = 0$ and choose distinct $a_{l+1}, \ldots, a_t$ in $\{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$ not equal to any of $a_1, \ldots, a_l$. (This means that we use $e(x) = \sum_{i=1}^{t} b_i x^{c_i}$ if $a_i = \beta^{c_i}$, which does not change the syndromes s_j as the new b_i are zero.) The outside matrices are again invertible by Proposition 5.1, so that l equals the rank of M_t. We also observe that the extended system

$$M'_t = \left(\begin{array}{ccccc|c} s_{m+1} & s_{m+2} & s_{m+3} & \cdots & s_{m+t} & s_{m+t+1} \\ s_{m+2} & s_{m+3} & s_{m+4} & \cdots & s_{m+t+1} & s_{m+t+2} \\ s_{m+3} & s_{m+4} & s_{m+5} & \cdots & s_{m+t+2} & s_{m+t+3} \\ \vdots & \vdots & \vdots & & & \vdots \\ s_{m+t} & s_{m+t+1} & s_{m+t+2} & \cdots & s_{m+2t-1} & s_{m+2t} \end{array}\right),$$

has a solution in F^t, given by the coefficients of $1, z, \ldots, z^{t-1}$ in $(a_1+z)(a_2+z)\ldots(a_t+z)$. (The calculations are identical to those for M'_l, we only used that $b_1, \ldots, b_l$ are non-zero in order to see that M_l is invertible.) Therefore M'_t is consistent, and its rank is l.

Note that for δ even, so $\delta = 2t+2$, the last syndrome σ_{m+2t+1} is never used in the system M'_t, and we discuss this case in Remark 3.10.

But for δ odd, so $\delta = 2t+1$, the calculations above gives rise to the following algorithm, which computes $e(x)$ with e of weight at most t such that $w(x) + e(x)$ is in the code, if any such $e(x)$ exists. Moreover, it will also detect when no such $e(x)$ exists. So, if $w(x)$ is not in the code, then the algorithm terminates in step 6 if and only if there exists $e(x)$ of positive weight at most equal to t that gives rise to the given syndromes $s_{m+1}, s_{m+2}, \ldots, s_{m+\delta-1}$, and it computes this as $\sum_{i=1}^{l} b_i x^{c_i}$ in the last step. (It follows from the calculations above that the algorithm computes $e(x)$ if it exists, and we discuss after Example 3.5 that if it terminates in step 6, then the $\tilde{e}(x) = \sum_{i=1}^{l} b_i x^{c_i}$ there is such that $w(x) + \tilde{e}(x)$ is in the code.)

Note that an $e(x)$ of weight at most t and such that $w(x)+e(x)$ is in the code, is automatically unique if it exists: for any $\tilde{e}(x)$ with those two properties $e(x)+\tilde{e}(x) = w(x)+e(x)+w(x)+\tilde{e}(x)$ is in the code and of weight at most $2t < \delta$, with δ the distance of the code, so $e(x) + \tilde{e}(x) = 0$.

Algorithm 3.3. Let the code be at the beginning of this section, with $\delta = 2t+1$ odd. Let $w(x)$ be a received polynomial.

1. Compute the syndromes $s_j = w(\beta^j)$ for $j = m+1, m+2, \ldots, m+2t$. If all those are 0 then STOP: $w(x)$ is in the code.

2. Write down the system M'_t. If it is inconsistent[2] then STOP: there are more than t errors.

3. Determine the rank l of the system M'_t. Write down the system M'_l in $(*)$ and solve it[3] to determine $\sigma(z) = \sigma_0 + \sigma_1 z + \cdots + \sigma_{l-1} z^{l-1} + z^l$.

4. Factorise $\sigma(z) = (a_1 + z)(a_2 + z)\ldots(a_l + z)$ for distinct a_i in $\{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$, and write $a_i = \beta^{c_i}$ for a unique c_i in $\{0, 1, \ldots, n-1\}$ in the process. If this is not possible then STOP: there are more than t errors.

5. Compute $b_1, b_2, \ldots, b_l$ from

$$\begin{pmatrix} a_1^{m+1} & a_2^{m+1} & \ldots & a_l^{m+1} \\ a_1^{m+2} & a_2^{m+2} & \ldots & a_l^{m+2} \\ \vdots & \vdots & & \vdots \\ a_1^{m+l} & a_2^{m+l} & \ldots & a_l^{m+l} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{pmatrix} = \begin{pmatrix} s_{m+1} \\ s_{m+2} \\ \vdots \\ s_{m+l} \end{pmatrix},$$

   which in extended matrix notation reads

$$(**) \qquad \left(\begin{array}{cccc|c} a_1^{m+1} & a_2^{m+1} & \ldots & a_l^{m+1} & s_{m+1} \\ a_1^{m+2} & a_2^{m+2} & \ldots & a_l^{m+2} & s_{m+2} \\ \vdots & \vdots & & \vdots & \vdots \\ a_1^{m+l} & a_2^{m+l} & \ldots & a_l^{m+l} & s_{m+l} \end{array}\right).$$

6. Decode $w(x)$ to $w(x) + \sum_{i=1}^{l} b_i x^{c_i}$.

Remark 3.4. Note that in step 5 of the algorithm there is a unique solution because of the properties of the Vandermonde matrices (see Proposition 5.1).

Example 3.5. Let $F = \{0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}$ with $\alpha^3 = 1 + \alpha$ and $\alpha^7 = 1$. We give some examples for a code over F of length 7 and with $\delta = 5$. We choose $m = -1$ and $\beta = \alpha^2$, so the generator polynomial is $g(x) = (1+x)(\beta + x)(\beta^2 + x)(\beta^3 + x)$ and the code has dimension 3. Note that $t = 2$, so in step 3 we find $\sigma(z) = \sigma_0 + \sigma_1 z + z^2$ if $l = 2$ and $\sigma(z) = \sigma_0 + z$ if $l = 1$.

(a) Let $s_0 = \alpha^3$, $s_1 = 0$, $s_2 = \alpha^6$ and $s_3 = \alpha^3$, so M'_2 is $\left(\begin{array}{cc|c} \alpha^3 & 0 & \alpha^6 \\ 0 & \alpha^6 & \alpha^3 \end{array}\right)$. Here $l = 2$, and solving this gives $\sigma(z) = \alpha^3 + \alpha^4 z + z^2$. In step 4 we find $\sigma(z) = (\beta + z)(\beta^4 + z)$, which is of the correct shape, with $a_1 = \beta$ and $a_2 = \beta^4$, so $c_1 = 1$ and $c_2 = 4$. In step 5 we solve $(**)$ for $l = 2$, i.e., we solve the system $\left(\begin{array}{cc|c} 1 & 1 & \alpha^3 \\ \alpha^2 & \alpha & 0 \end{array}\right)$ for b_1 and b_2 because $a_1 = \beta = \alpha^2$ and $a_2 = \beta^4 = \alpha$. This gives $b_1 = 1$ and $b_2 = \alpha$, so in step 6 we decode $w(x)$ to $w(x) + x + \alpha x^4$.

[2]This means there is no solution, which can be decided by performing Gauss elimination on the rows.

[3]By Proposition 5.2 there is always a unique solution.

(b) Let $s_0 = 1$, $s_1 = \alpha$, $s_2 = \alpha^5$ and $s_3 = \alpha^6$, so M_2' is $\left(\begin{array}{cc|c} 1 & \alpha & \alpha^5 \\ \alpha & \alpha^5 & \alpha^6 \end{array}\right)$. Here $l = 2$, and solving this gives $\sigma(z) = \alpha^5 + z^2$. In step 4 we find $\sigma(z) = (\beta^3 + z)(\beta^3 + z)$, which is not of the correct shape as not all roots are distinct, so we conclude there that there are more than 2 errors.

(c) Let $s_0 = \alpha$, $s_1 = 1$, $s_2 = \alpha^6$ and $\sigma_3 = \alpha^5$, so M_2' is $\left(\begin{array}{cc|c} \alpha & 1 & \alpha^6 \\ 1 & \alpha^6 & \alpha^5 \end{array}\right)$. Here $l = 1$, because the second equation is α^6 times the first. So we use M_1', which is $\left(\alpha \,\middle|\, 1\right)$, so that $\sigma(z) = \alpha^6 + z$. In step 4 we write this as $\sigma(z) = \beta^3 + z$, so that $c_1 = 3$. In step 5 we solve $(**)$ for $l = 1$, i.e., $\left(1 \,\middle|\, \alpha\right)$, so that $b_1 = \alpha$. In step 6 we decode $w(x)$ to $w(x) + \alpha x^3$.

(d) Let $s_0 = 1$, $s_1 = s_2 = s_3 = 0$, so M_2' is $\left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right)$. Here $l = 1$, so we use M_1', which is $\left(1 \,\middle|\, 0\right)$. So $\sigma(z) = z$, and in step 4 we conclude that there are more than 2 errors.

(e) Let $s_0 = 1$, $s_1 = \alpha$, $s_2 = 0$ and $s_3 = 1$, so M_2' is $\left(\begin{array}{cc|c} 1 & \alpha & 0 \\ \alpha & 0 & 1 \end{array}\right)$. Here $l = 2$, and solving this gives $\sigma(z) = \alpha^6 + \alpha^5 z + z^2$, and in step 4 we conclude that there are more than 2 errors because $\sigma(z)$ does not have two distinct roots among $1, \beta, \beta^2, \ldots, \beta^6$ (it has no roots among those at all).

We have already seen that Algorithm 3.3 computes $e(x)$ if $w(x) = v(x) + e(x)$ with $v(x)$ in the code and e of weight at most t. We still have to see that if the algorithm terminates in step 6 then the resulting word is in the code, which is where we shall use that δ is odd because s_i for $i = m+1, m+2, \ldots, m+2t$ are then all syndromes. In order to avoid a proof with many abstract indices, we prove this in an example.

Example 3.6. Let us look at what happens if the algorithm terminates in step 6 for $t = 4$ and $l = 2$. So in step 2 we have the system

$$M_4' = \left(\begin{array}{cccc|c} s_{m+1} & s_{m+2} & s_{m+3} & s_{m+4} & s_{m+5} \\ s_{m+2} & s_{m+3} & s_{m+4} & s_{m+5} & s_{m+6} \\ s_{m+3} & s_{m+4} & s_{m+5} & s_{m+6} & s_{m+7} \\ s_{m+4} & s_{m+5} & s_{m+6} & s_{m+7} & s_{m+8} \end{array}\right),$$

and we are assuming that the system is consistent, so we move on to step 3.

We compute that the rank is 2 (but this is probably done while checking the system for consistency already). Then by Proposition 5.2, the first two rows are independent, and the last two rows depend on those. In fact, by that proposition, the submatrix

$$\begin{pmatrix} s_{m+1} & s_{m+2} \\ s_{m+2} & s_{m+3} \end{pmatrix}$$

is invertible. The algorithm prescribes we now solve the extended system

$$M_2' = \left(\begin{array}{cc|c} s_{m+1} & s_{m+2} & s_{m+3} \\ s_{m+2} & s_{m+3} & s_{m+4} \end{array}\right),$$

which has a unique solution, so we have

$$\begin{pmatrix} s_{m+1} & s_{m+2} \\ s_{m+2} & s_{m+3} \end{pmatrix} \begin{pmatrix} \sigma_0 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} s_{m+3} \\ s_{m+4} \end{pmatrix}$$

for a unique pair (σ_0, σ_1). We then form the polynomial $\sigma(z) = \sigma_0 + \sigma_1 z + z^2$, which ends step 3.

In step 4 we factorise this as $\sigma(z) = (a_1 + z)(a_2 + z)$ with $a_1 = \beta^{c_1}$ and $a_2 = \beta^{c_2}$ for c_1 and c_2 distinct in $\{0, 1, 2, \ldots, n-1\}$. In step 5 we then solve

$$\begin{pmatrix} a_1^{m+1} & a_2^{m+1} \\ a_1^{m+2} & a_2^{m+2} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} s_{m+1} \\ s_{m+2} \end{pmatrix},$$

which has a unique solution because $a_1 \neq a_2$ and both are non-zero. In step 6 we correct $w(x)$ by adding $\tilde{e}(x) = b_1 x^{c_1} + b_2 x^{c_2}$ to it.

Let us verify directly from the computations in the algorithm that the result is actually a codeword. If $\tilde{s}_i = \tilde{e}(\beta^i)$ for $i = m+1, \ldots, m+8$ are the syndromes of $\tilde{e}(x)$, then we have to check that $\tilde{s}_i = s_i$ for those i. From the last matrix equation that was solved, we know that $\tilde{s}_{m+1} = \tilde{e}(\beta^{m+1}) = b_1(\beta^{m+1})^{c_1} + b_2(\beta^{m+1})^{c_2} = b_1 a_1^{m+1} + b_2 a_2^{m+1} = s_{m+1}$, and a similar calculation shows that $\tilde{s}_{m+2} = s_{m+2}$. From the calculations at the beginning of this section, applied to $\tilde{e}(x)$, we know that

$$\begin{pmatrix} \tilde{s}_{m+1} & \tilde{s}_{m+2} \\ \tilde{s}_{m+2} & \tilde{s}_{m+3} \end{pmatrix} \begin{pmatrix} \sigma_0 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} \tilde{s}_{m+3} \\ \tilde{s}_{m+4} \end{pmatrix} \tag{3.7}$$

because $(a_1 + z)(a_2 + z) = \sigma(z)$ is the error locator polynomial for $\tilde{e}(x)$. But we also know that

$$\begin{pmatrix} s_{m+1} & s_{m+2} \\ s_{m+2} & s_{m+3} \end{pmatrix} \begin{pmatrix} \sigma_0 \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} s_{m+3} \\ s_{m+4} \end{pmatrix} \tag{3.8}$$

as this is the identity from which we computed σ_0 and σ_1. From $\tilde{s}_{m+1} = s_{m+1}$ and $\tilde{s}_{m+2} = s_{m+2}$, we find, using both equations, that $\tilde{s}_{m+3} = \tilde{s}_{m+1}\sigma_0 + \tilde{s}_{m+2}\sigma_1 = s_{m+1}\sigma_0 + s_{m+2}\sigma_1 = s_{m+3}$. But then also $\tilde{s}_{m+4} = \tilde{s}_{m+2}\sigma_0 + \tilde{s}_{m+3}\sigma_1 = s_{m+2}\sigma_0 + s_{m+3}\sigma_1 = s_{m+4}$.

Let us now prove that $b_1 \neq 0$ and $b_2 \neq 0$. From the calculations at the beginning of this section applied to $\tilde{e}(x)$ we know that the weight of $\tilde{e}$ equals the rank of the 2×2 matrix in (3.7). But we now know that this is identical to the 2×2 matrix in (3.8), which has rank 2 as we noticed before. So $\tilde{e}$ has weight 2, hence b_1 and b_2 are both non-zero as the other positions in $\tilde{e}$ are zero.

It remains to show that $\tilde{s}_i = s_i$ for $i = m+5$, $m+6$, $m+7$ and $m+8$. For this, we compare

$$M_4' = \left(\begin{array}{cccc|c} s_{m+1} & s_{m+2} & s_{m+3} & s_{m+4} & s_{m+5} \\ s_{m+2} & s_{m+3} & s_{m+4} & s_{m+5} & s_{m+6} \\ s_{m+3} & s_{m+4} & s_{m+5} & s_{m+6} & s_{m+7} \\ s_{m+4} & s_{m+5} & s_{m+6} & s_{m+7} & s_{m+8} \end{array}\right)$$

and

$$\widetilde{M_4'} = \left(\begin{array}{cccc|c} s_{m+1} & s_{m+2} & s_{m+3} & s_{m+4} & \tilde{s}_{m+5} \\ s_{m+2} & s_{m+3} & s_{m+4} & \tilde{s}_{m+5} & \tilde{s}_{m+6} \\ s_{m+3} & s_{m+4} & \tilde{s}_{m+5} & \tilde{s}_{m+6} & \tilde{s}_{m+7} \\ s_{m+4} & \tilde{s}_{m+5} & \tilde{s}_{m+6} & \tilde{s}_{m+7} & \tilde{s}_{m+8} \end{array}\right),$$

where in the latter we used that $\tilde{s}_i = s_i$ for $i = m+1$, $m+2$, $m+3$, and $m+4$. Both have rank 2: the former by our assumptions in step 3 of the algorithm, the latter because the calculations at the beginning of this section applied to $\tilde{e}(x)$ show the rank to equal the weight of $\tilde{e}$, which we now know is 2. So there are d_1 and d_2 in F such that the third row of M_4' is the sum of d_1 times the first and d_2 times the second row. Because the 2×2 matrix in the upper lefthand corner of M_4' is invertible, the d_1 and d_2 are unique.

The same statements apply to the rows of $\widetilde{M_4'}$ with potentially different $\tilde{d}_1$ and $\tilde{d}_2$. But (d_1, d_2) and $(\tilde{d}_1, \tilde{d}_2)$ are determined entirely by the first two positions in the rows considered, and here

everything coincides. So $\tilde{d}_1 = d_1$ and $\tilde{d}_2 = d_2$. Then from the third position in the third rows of $\widetilde{M}'_4$ and M'_4, we see $\tilde{s}_{m+5} = d_1 s_{m+3} + d_2 s_{m+4} = s_{m+5}$. Next, considering the fourth position, we see similarly that $\tilde{s}_{m+6} = s_{m+6}$, and then from the fifth position that $\tilde{s}_{m+7} = s_{m+7}$.

We can then write the fourth rows of $\tilde{M}'_4$ and M'_4 as a linear combination of the first two rows of those matrices, but those first two rows are now identical, and the linear combination is also identical as this is determined entirely by the first two positions everywhere. It then follows by a similar calculation that $\tilde{s}_{m+8} = s_{m+8}$.

So all syndromes of $w(x)$ and $\tilde{e}(x)$ coincide, hence the element $w(x) + \tilde{e}(x)$ computed by the algorithm in step 6 is in the code.

Exercise 3.9. Modify the example above to prove in general that Algorithm 3.3, if it terminates in step 6, computes an element of the code.

Remark 3.10. In Algorithm 3.3, the assumption that δ is odd, so that $\delta = 2t + 1$, is used first in step 2 because if $w(x)$ is not in the code, then at least one s_i for $i = m+1, m+2, \ldots, m+2t$ is non-zero as these are all syndromes. It is also used in the proof above that, if the algorithm terminates with step 6, then the resulting $w(x) + \sum_{i=1}^{l} b_i c^{c_i}$ is in the code, again because the algorithm ensures that the corresponding syndromes s_i are zero for $i = m+1, \ldots, m+2t$, and those are all syndromes.

If δ is even, so $\delta = 2t + 2$, then this has to be modified as follows, because in this case the algorithm does not use the last syndrome s_{2t+1} at all. We compare everything with the larger code with generator polynomial $g(x) = (\beta^{m+1} + x)(\beta^{m+2} + x) \ldots (\beta^{m+\delta-2} + x)$, which is one dimension larger but has the same t as its distance is $2t + 1$.

In step 1, we have to avoid going into step 2 with $s_i = 0$ for $i = m+1, m+2, \ldots, m+2t$. For this, we include the statement that if this holds for $w(x)$ but $s_{m+2t+1} \neq 0$, then there are too many errors. To see this, note that if $w(x)$ is at distance at most t from a (unique) codeword in the original code, then the same is true for the larger code, with the same codeword as t is the same for both codes. But here that codeword is $w(x)$ itself, which is not in the original code.

The same applies if the algorithm terminates in step 6. Now the word $w(x) + \sum_{i=1}^{l} b_i x^{c_i}$ has syndromes $s_i = 0$ for $i = m+1, \ldots, m+2t$, but we do not know that $s_{m+2t+1} = 0$. If $s_{m+2t+1} = 0$, then we corrected the error because we found a codeword in the original code at distance $l \leq t$ from $w(x)$. If $s_{2m+t+1} \neq 0$, then the same argument applies as when modifying step 1: if $w(x)$ is at distance at most t from a codeword in the original code, then the algorithm would have computed that codeword in the larger code. As the result is not in the original code, in this case we conclude there are more than t errors.

4 Decoding Reed-Solomon codes: the transform method

In this section, we discuss a decoding method for decoding $RS(n, \delta)$ that works the same way for δ odd and δ even.

As in the previous two sections, we fix the following notation. We have a field $F = GF(2^r)$ for $r \geq 2$, an n that divides $2^r - 1$, and a non-zero β in F with $1, \beta, \beta^2, \beta^3, \ldots, \beta^{n-1}$ all distinct as well as $\beta^n = 1$.

To every polynomial $v(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}$ in $F[x]$ we associate the polynomial $V(z) = V_0 + V_1 z + \cdots + V_{n-1} z^{n-1}$ in $F[z]$ with $V_i = v(\beta^{-i})$ for $i = 0, 1, \ldots, n-1$, and to every polynomial $V(z) = V_0 + V_1 z + \cdots + V_{n-1} z^{n-1}$ in $F[z]$ we associate the polynomial $v(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}$ in $F[x]$ with $v_i = V(\beta^i)$ for $i = 0, 1, \ldots, n-1$. (Note that we use β^{-i} in the first map but β^i in the second.)

Proposition 4.1. The maps $v(x) \mapsto V(z)$ and $V(z) \mapsto v(x)$ are linear, and inverse to each other.

Proof. Let

$$A = (\beta^{ij})_{i,j=0,\dots,n-1} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \beta & \beta^2 & \dots & \beta^{n-1} \\ 1 & \beta^2 & \beta^4 & \dots & \beta^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \beta^{n-1} & \beta^{2(n-1)} & \dots & \beta^{(n-1)^2} \end{pmatrix}.$$

Then A is invertible, with inverse $A^{-1} = (\beta^{-ij})_{i,j=0,\dots,n-1}$: the entry in the product $(\beta^{ij})(\beta^{-ij})$ at position (i,j) is $\sum_{k=0}^{n-1} \beta^{ik}\beta^{-kj} = \sum_{k=0}^{n-1}(\beta^{i-j})^k$, which equals 1 if $i = j$ because n is odd, and equals $(1+\beta^{i-j})^{-1}(1+(\beta^{i-j})^n) = 0$ otherwise. Moreover, directly from the definitions one obtains that $(v_0, v_1, \dots, v_{n-1}) = (V_0, V_1, \dots, V_{n-1})A$ and $(V_0, V_1, \dots, V_{n-1}) = (v_0, v_1, \dots, v_{n-1})A^{-1}$. This shows that the maps are linear, and inverse to each other. □

We now fix m in $\mathbb{Z}$ and δ with $2 \le \delta \le n$ and consider the Reed-Solomon code $RS(n,\delta)$ of length n with generator polynomial $g(x) = (\beta^{m+1}+x)(\beta^{m+2}+x)\dots(\beta^{m+\delta-1}+x)$, as in Section 3. Suppose $w(x)$ in $F[x]$ is of degree less than n, and equals $v(x)+e(x)$ with both $v(x)$ and $e(x)$ in $F[x]$ of degree less than n, $v(x)$ in the code, and e of weight at most $t = \lfloor \frac{\delta-1}{2} \rfloor$. Then

$$s_j = w(\beta^j) = v(\beta^j) + e(\beta^j) = e(\beta^j) = E_{-j}$$

for $j = m+1, m+2, \dots, m+\delta-1$. Here, and below, we consider the indices of the E_k modulo n, which is possible because $\beta^n = 1$. We assume that $w(x)$ is not in the code, so some s_j is non-zero, as is $e(x)$. If $l = \mathrm{wt}(e)$, then we have $e(x) = e_0 + e_1 x + \dots + e_{n-1}x^{n-1} = \sum_{i=1}^{l} b_i x^{c_i}$ as at the beginning of Section 3, with all b_i non-zero and all c_i distinct in $\{0, 1, \dots, n-1\}$. We then form the error locator polynomial $\sigma(z) = \prod_{i=1}^{l}(\beta^{c_i}+z)$ in $F[x]$ as before. For $k = 0, 1, \dots, n-1$ we have that $\sigma(\beta^k) \neq 0$ implies that $0 = e_k = E(\beta^k)$, so that $\sigma(z)E(z)$ has $1, \beta, \beta^2, \beta^3, \dots, \beta^{n-1}$ among its roots. We have $1+z^n = (1+z)(\beta+z)(\beta^2+z)\dots(\beta^{n-1}+z)$ by Lemma 2.1, so that $\sigma(z)E(z)$ is divisible by $1+z^n$ in $F[z]$. In other words

$$(***) \qquad \sigma(z)E(z) \equiv 0 \text{ modulo } 1+z^n\,.$$

Here we know $E_{-j} = s_j$ for $j = m+1, m+2, \dots, m+\delta-1$.

Conversely, if $\sigma(z)$ is any monic polynomial in $F[z]$ of degree at most t for which $(***)$ holds, for some $E(z)$ in $F[z]$ of degree less than n with $E_{-j} = s_j$ for $j = m+1, m+2, \dots, m+\delta-1$, then the corresponding $e(x)$ satisfies $e(\beta^j) = E_{-j} = s_j$ for those j, so $w(x)+e(x)$ is in the code. Also, $\sigma(z)E(z)$ has $1, \beta, \beta^2, \beta^3, \dots, \beta^{n-1}$ among its roots, and at most t of those can be roots of $\sigma(z)$. So at least $n-t$ of those are roots of $E(z)$. That means that at least $n-t$ of the $e_k = E(\beta^k)$ are zero, so $\mathrm{wt}(e) \le t$. Hence we decoded $w(x)$ to $w(x)+e(x)$ in the code by fixing at most t errors.

We now have proved the following theorem.

Theorem 4.2. For $w(x)$ not in the code, with syndromes s_j for $j = m+1, m+2, \dots, m+\delta-1$, the following are equivalent:

(a) $w(x)$ can be corrected to a codeword by fixing at most t errors;

(b) there exist $\sigma(z)$ and $E(z)$ in $F[z]$ satisfying $(***)$ with $\sigma(z)$ monic and of degree at most t, $E(z)$ of degree less than n, and $E_{-j} = s_j$ for $j = m+1, m+2, \dots, m+\delta-1$.

The correction is then carried out by adding $e(x) = \sum_{k=0}^{n-1} E(\beta^k)x^k$ to $w(x)$.

Remark 4.3. (a) The $E(z)$ in any solution as in Theorem 4.2(b) is unique, even though $\sigma(z)$ may not be (if the actual error locator polynomial has degree less than t, we can multiply it with monic factors of degree $t - \deg(\sigma(z))$). To see this, suppose $E(z)$ and $\tilde{E}(z)$ are both parts of solutions. Then $e(x) + \tilde{e}(x)$ has weight at most $2t$ and is in the code because the syndromes are zero, hence is identically 0. So $\tilde{e}(x) = e(x)$, and therefore $\tilde{E}(z) = E(z)$ by Proposition 4.1.

(b) Note that if $\sigma(\beta^k) \neq 0$ then $E(\beta^k) = 0$, which cuts down on the amount of calculation needed because $\deg(\sigma(z)) \leq t \leq \frac{n-1}{2}$ (and it is in practice often much smaller than this), so that $\sigma(z)$ is of (much) lower degree than $E(z)$. So it is in general less work to evaluate $\sigma(z)$ for $1, \beta, \beta^2, \ldots, \beta^{n-1}$ and then evaluate $E(\beta^k)$ for those (at most t) values of k for which $\sigma(\beta^k) \neq 0$, than to compute $E(\beta^k)$ for $k = 0, 1, \ldots, n-1$.

We now try to decode $w(x)$ by using Theorem 4.2(b). For this, we let

$$\sigma(z) = \sigma_0 + \sigma_1 z + \cdots + \sigma_{t-1} z^{t-1} + z^t$$
$$E(z) = E_0 + E_1 z + \cdots + E_{n-1} z^{n-1}$$

with $\sigma_0, \ldots, \sigma_{t-1}$ variables, $E_{-j} = s_j$ for $j = m+1, m+2, \ldots, m+\delta-1$ (indices modulo n as always), and the other E_k variables. Considering the coefficients of z^k for $k = 0, 1, \ldots, n-1$, we see that $(***)$ is equivalent to a linear system with n equations:

$$\sigma_0 E_k + \sigma_1 E_{k-1} + \cdots + \sigma_{t-1} E_{k-t+1} = E_{k-t} \qquad (k = 0, 1, \ldots, n-1). \tag{4.4}$$

Note that a term $\sigma_i z^i E_j z^j = \sigma_i E_j z^{i+j}$ for $i = 0, 1, \ldots, t-1$ and $j = 0, 1, \ldots, n-1$ modulo $1+z^n$ remains the same if $i + j < n$, but is replaced with $\sigma_i E_j z^{i+j-n}$ if $i + j \geq n$. So it contributes to the coefficient z^k in z^k in $\sigma(z)E(z)$ modulo $1 + z^k$ for the unique $k = 0, 1, \ldots, n-1$ with $i + j \equiv k$ modulo n. Similarly, the term $z^t E_j z^j = E_j z^{t+j}$ contributes to the unique k with $t + j \equiv k$ modulo n. So for a fixed $k = 0, 1, \ldots, n-1$, we find as coefficient of z^k in $\sigma(z)E(z)$ modulo $1 + z^n$ the sum of those $\sigma_i E_j$ with $i + j \equiv k$ modulo n, and also E_j with $t + j \equiv k$ modulo n. This is exactly what is stated in (4.4).

Example 4.5. Let us take $n = 7$ and $\delta = 5$, so $t = 2$. Then the system is

$$\begin{aligned}
\sigma_0 E_0 + \sigma_1 E_6 &= E_5 \\
\sigma_0 E_1 + \sigma_1 E_0 &= E_6 \\
\sigma_0 E_2 + \sigma_1 E_1 &= E_0 \\
\sigma_0 E_3 + \sigma_1 E_2 &= E_1 \\
\sigma_0 E_4 + \sigma_1 E_3 &= E_2 \\
\sigma_0 E_5 + \sigma_1 E_4 &= E_3 \\
\sigma_0 E_6 + \sigma_1 E_5 &= E_4 \,.
\end{aligned}$$

where we listed the equations in the order corresponding to the coefficients of $1, z, z^2, \ldots, z^6$ in $\sigma(z)E(z)$ modulo $1 + z^7$. Here $\sigma_i E_j$ comes from a term $\sigma_i E_j z^{i+j}$ that was replaced with $\sigma_i E_j z^{i+j-7}$ when $i + j \geq 7$, which is only the case for $i = 1$ and $j = 6$. Similarly, E_j on the righthand side comes from a term $E_j z^{2+j}$ that was replaced with $E_j z^{j-7}$ when $2 + j \geq 7$, which is the case for $j = 5$ and 6.

Note that (4.4) is an inhomogeneous system of n linear equations in the variables $\sigma_0, \ldots, \sigma_{t-1}$, with as coefficients $E_0, E_1, \ldots, E_{n-1}$, some of which are known, and some of which are variables. But we know all the relevant E_j for a subset of these equations, namely where $k-t, \ldots, k$ are all among $-m-\delta+1, \ldots, -m-1$, because $E_{-j} = s_j$ for $j = m+1, m+2, \ldots, m+\delta-1$. This holds

in the range from where $k-t=-m-\delta+1$, to where $k=-m-1$, i.e., from $k=-m+t-\delta+1$ to $k=-m-1$ (again indices modulo n). Those are $\delta-t-1$ consecutive equations. Because $\delta=2t+1$ or $2t+2$, this includes the range $k=-m-t,\ldots,-m-1$. The equations in this range are then also

$$\sigma_0 s_{-k}+\sigma_1 s_{1-k}+\cdots+\sigma_{t-1}s_{t-1-k}=s_{t-k}$$

and they give us precisely the extended system

$$M'_t=\left(\begin{array}{ccccc|c} s_{m+1} & s_{m+2} & s_{m+3} & \cdots & s_{m+t} & s_{m+t+1} \\ s_{m+2} & s_{m+3} & s_{m+4} & \cdots & s_{m+t+1} & s_{m+t+2} \\ s_{m+3} & s_{m+4} & s_{m+5} & \cdots & s_{m+t+2} & s_{m+t+3} \\ \vdots & \vdots & \vdots & & & \vdots \\ s_{m+t} & s_{m+t+1} & s_{m+t+2} & \cdots & s_{m+2t-1} & s_{m+2t} \end{array}\right)$$

that we encountered in Section 3.

Assume that there is some solution to (∗∗∗) as in Theorem 4.2(b). Then there must be a solution to this system M'_t as well. Also, by Theorem 4.2, $w(x)+e(x)$ is in the code if $E(z)$ (with some values for all unknown E_k) is part of a solution of (∗∗∗), so the calculations at the beginning of Section 3 apply (these did not use that δ was odd). In particular, the rank l of this system is the weight of the error e. Replacing m with $m+2t-2l$ in (3.2), we see that the $l\times l$ submatrix in the bottom righthand corner of

$$M_t=\begin{pmatrix} s_{m+1} & s_{m+2} & s_{m+3} & \cdots & s_{m+t} \\ s_{m+2} & s_{m+3} & s_{m+4} & \cdots & s_{m+t+1} \\ s_{m+3} & s_{m+4} & s_{m+5} & \cdots & s_{m+t+2} \\ \vdots & \vdots & \vdots & & \\ s_{m+t} & s_{m+t+1} & s_{m+t+2} & \cdots & s_{m+2t-1} \end{pmatrix}$$

is invertible. We know that if we let $\sigma(z)$ be the product of z^{t-l} and the error locator polynomial, and $E(z)$ the transform of $e(x)$, then $\sigma(z)$ and $E(z)$ give a solution to (∗∗∗). So there is a solution with $\sigma(z)$ having $\sigma_0=\sigma_1=\cdots=\sigma_{t-l-1}=0$, and because of the $l\times l$ invertible submatrix of M_t we have just described, there is a unique such solution.

So if there is any solution to (∗∗∗) at all, with the conditions on $\sigma(z)$ and $E(z)$ as before, then there exists a unique one with $\sigma_0=\sigma_1=\cdots=\sigma_{t-l-1}=0$ for l the rank of M'_t, and $w(x)+e(x)$ is in the code for $e(x)$ the transform of the $E(z)$ in this solution.

We therefore have the following algorithm.

Algorithm 4.6. Let the code be as at the beginning of this section, with $t=\lfloor\frac{\delta-1}{2}\rfloor$, and let $w(x)$ be a received polynomial.

1. Compute the syndromes $s_j=w(\beta^j)$ for $j=m+1,m+2,\ldots,m+\delta-1$. If all those are 0 then STOP: $w(x)$ is in the code.

2. Write down the system (∗∗∗) in its equivalent form (4.4).

3. In here find the t equations corresponding to M'_t. If this is inconsistent then STOP: there are more than t errors. If it is consistent, let l be its rank.

4. Put $\sigma_0=\sigma_1=\cdots=\sigma_{t-l-1}=0$ in (4.4), and solve the equations corresponding to M'_t for $\sigma_{t-l},\ldots,\sigma_{t-1}$. If there is not precisely one solution then STOP: there are more than t errors.

5. Substitute those values in the remaining $n-t$ equations in (4.4), which results in a system of $n-t$ equations in the $n-\delta+1$ unknown E_k.

6. If there is no solution to this system then STOP: there are more than t errors. If there is a solution, decode $w(x)$ to $w(x)+\sum_{i=0}^{n-1} E(\beta^i)x^i$.

Example 4.7. We redo the examples in Example 3.5 using this method. Recall that in those examples we had $F=\{0,1,\alpha,\alpha^2,\alpha^3,\alpha^4,\alpha^5,\alpha^6\}$ with $\alpha^3=1+\alpha$ and $\alpha^7=1$, $\beta=\alpha^2$, length 7, $\delta=5$, $m=-1$, and generator polynomial $g(x)=(1+x)(\beta+x)(\beta^2+x)(\beta^3+x)$. As $t=2$, $(***)$ results in the 7 equations $\sigma_0 E_k+\sigma_1 E_{k-1}=E_{k-2}$ for $k=0,1,\ldots,6$ in σ_0 and σ_1, with $E_{-j}=s_j$ (indices modulo 7) known for $j=0$, 1, 2 and 3, which we indicated using bold:

$$\begin{aligned}
\sigma_0\mathbf{E}_0+\sigma_1\mathbf{E}_6&=\mathbf{E}_5\\
\sigma_0 E_1+\sigma_1\mathbf{E}_0&=\mathbf{E}_6\\
\sigma_0 E_2+\sigma_1 E_1&=\mathbf{E}_0\\
\sigma_0 E_3+\sigma_1 E_2&=E_1\\
\sigma_0\mathbf{E}_4+\sigma_1 E_3&=E_2\\
\sigma_0\mathbf{E}_5+\sigma_1\mathbf{E}_4&=E_3\\
\sigma_0\mathbf{E}_6+\sigma_1\mathbf{E}_5&=\mathbf{E}_4\,.
\end{aligned}$$

Here we listed them in the order $k=0,1,\ldots,6$, and the system $(*)$ or M_2' corresponds to the equations for $k=0$ and 6. After using it to determine σ_0 and σ_1 (if a solution of M_2' exists) we can then use the equation for $k=5$ to compute E_3, then the equation for $k=4$ to compute E_2, and the equation for $k=3$ to compute E_1. At that stage all E_j are known but the equations for $k=2$ and $k=1$ still have to be checked, as they might rule out the existence of any solution of the full system (which would show that $w(x)$ cannot be decoded using at most 2 errors).

We now carry out Algorithm 4.6 for the examples in Example 3.5 but starting at step 4, using the outcome of step 3 from the calculations done in Example 3.5. (Step 2 has just been carried out in general for this code.)

(a) We have $E_0=s_0=\alpha^3$, $E_6=s_1=0$, $E_5=s_2=\alpha^6$ and $E_4=s_3=\alpha^3$, and had already found from M_2' that $l=2$, $\sigma_0=\alpha^3$ and $\sigma_1=\alpha^4$. We then find

$$\begin{aligned}
E_3&=\alpha^3E_5+\alpha^4E_4=\alpha^9+\alpha^7=\alpha^6\\
E_2&=\alpha^3E_4+\alpha^4E_3=\alpha^6+\alpha^{10}=\alpha^4\\
E_1&=\alpha^3E_3+\alpha^4E_2=\alpha^9+\alpha^8=\alpha^4
\end{aligned}$$

and end up with two check equations

$$\begin{aligned}
E_0&=\alpha^3E_2+\alpha^4E_1\\
E_6&=\alpha^3E_1+\alpha^4E_0,
\end{aligned}$$

which are satisfied because $\alpha^3=\alpha^3\alpha^4+\alpha^4\alpha^4$ and $0=\alpha^3\alpha^4+\alpha^4\alpha^3$. So we reach step 6 with $\sigma(z)=\alpha^3+\alpha^4z+z^2$ and $E(z)=\alpha^3+\alpha^4z+\alpha^4z^2+\alpha^6z^3+\alpha^3z^4+\alpha^6z^5$ (as $E_6=0$). Then $\sigma(\beta^k)\neq 0$ for $k=0,1,\ldots,6$ only when $k=1$ or 4, so $e(x)=E(\beta)x+E(\beta^4)x^4=x+\alpha x^4$.

(b) We have $E_0=s_0=1$, $E_6=s_1=\alpha$, $E_5=s_2=\alpha^5$ and $E_4=s_3=\alpha^6$. We have found from M_2' that $l=2$, $\sigma_0=\alpha^5$ and $\sigma_1=0$. We then find

$$\begin{aligned}
E_3&=\alpha^5E_5=\alpha^{10}=\alpha^3\\
E_2&=\alpha^5E_4=\alpha^{11}=\alpha^4\\
E_1&=\alpha^5E_3=\alpha^8
\end{aligned}$$

and end up with two check equations

$$E_0 = \alpha^5 E_2$$
$$E_6 = \alpha^5 E_1,$$

which read $1 = \alpha^5\alpha^4$ and $\alpha = \alpha^5\alpha^8$, so both fail. So (4.4) has no solution, and we conclude there that there are more than 2 errors.

(c) We have $E_0 = s_0 = \alpha$, $E_6 = s_1 = 1$, $E_5 = s_2 = \alpha^6$ and $E_4 = \sigma_3 = \alpha^5$. We have that M_2' is $\left(\begin{array}{cc|c} \alpha & 1 & \alpha^6 \\ 1 & \alpha^6 & \alpha^5 \end{array}\right)$ with $l = 1$, so we put $\sigma_0 = 0$. This results in a unique solution, with $\sigma_1 = \alpha^6$. We then find

$$E_3 = \alpha^6 E_4 = \alpha^{11} = \alpha^4$$
$$E_2 = \alpha^6 E_3 = \alpha^{10} = \alpha^3$$
$$E_1 = \alpha^6 E_2 = \alpha^9 = \alpha^2$$

and end up with two check equations

$$E_0 = \alpha^6 E_1$$
$$E_6 = \alpha^6 E_0,$$

which read $\alpha = \alpha^6\alpha^2$ and $1 = \alpha^6\alpha$, so both are satisfied. We therefore reach step 6 with $\sigma(z) = \alpha^6 z + z^2$ and $E(z) = \alpha + \alpha^2 z + \alpha^3 z^2 + \alpha^4 z^3 + \alpha^5 z^4 + \alpha^6 z^5 + z^6$. Then $\sigma(\beta^k) \neq 0$ for $k = 0, 1, \ldots, 6$ only when $k = 3$, so $e(x) = E(\beta^3)x^3 = \alpha x^3$.

(d) We have $E_0 = s_0 = 1$, $E_6 = s_1 = 0$, $E_5 = s_2 = 0$ and $E_4 = s_3 = 0$, so M_2' is $\left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right)$. Here $l = 1$, so we use $\sigma_0 = 0$. But then there is no unique solution to this as σ_1 is undetermined. So we conclude that there are more than 2 errors.

(e) We have $E_0 = s_0 = 1$, $E_6 = s_1 = \alpha$, $E_5 = s_2 = 0$ and $E_4 = s_3 = 1$. We had already seen that $l = 2$ and that we have a unique solution $\sigma_0 = \alpha^6$ and $\sigma_1 = \alpha^5$ to M_2'.

$$E_3 = \alpha^6 E_5 + \alpha^5 E_4 = 0 + \alpha^5 = \alpha^5$$
$$E_2 = \alpha^6 E_4 + \alpha^5 E_3 = \alpha^6 + \alpha^{10} = \alpha^4$$
$$E_1 = \alpha^6 E_3 + \alpha^5 E_2 = \alpha^{11} + \alpha^9 = \alpha$$

and end up with two check equations

$$E_0 = \alpha^6 E_2 + \alpha^5 E_1$$
$$E_6 = \alpha^6 E_1 + \alpha^5 E_0,$$

which read $1 = \alpha^6\alpha^4 + \alpha^5\alpha$ and $\alpha = \alpha^6\alpha + \alpha^4 \cdot 1$. Both fail, so we conclude that there are more than 2 errors.

5 Appendix: some linear algebra

Proposition 5.1. Let $t \geq 1$, and let $\alpha_1, \ldots, \alpha_t$ be distinct elements in any field F. Then the rows (and the columns) of

$$\begin{pmatrix} 1 & \alpha_1 & \ldots & \alpha_1^{t-1} \\ 1 & \alpha_2 & \ldots & \alpha_2^{t-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_t & \ldots & \alpha_t^{t-1} \end{pmatrix}$$

are linearly independent over F. If $\alpha_1, \ldots, \alpha_t$ are also non-zero, and m is any integer, then the same holds for the rows (and the columns) of

$$\begin{pmatrix} \alpha_1^{m+1} & \alpha_1^{m+2} & \ldots & \alpha_1^{m+t} \\ \alpha_2^{m+1} & \alpha_2^{m+2} & \ldots & \alpha_2^{m+t} \\ \vdots & \vdots & & \vdots \\ \alpha_t^{m+1} & \alpha_t^{m+2} & \ldots & \alpha_t^{m+t} \end{pmatrix}.$$

Proof. The first matrix is known as a Vandermonde matrix, and the statement about its rows and columns follows from the fact that its determinant is non-zero as that determinant is equal to $\prod_{1 \le i < j \le t}(\alpha_j - \alpha_i)$. This formula is proved in any decent book on linear algebra (and can be proved by induction on t). The second statement follows because the determinant of this matrix equals $(\alpha_1 \alpha_2 \ldots \alpha_t)^{m+1}$ times the determinant of the first matrix: take out a factor α_i^{m+1} from the ith row. □

Proposition 5.2. Let F be a field, $a, b \ge 0$, and

$$\left(\begin{array}{ccccc|c} s_0 & s_1 & s_2 & \ldots & s_a & s_{a+1} \\ s_1 & s_2 & s_3 & \ldots & s_{a+1} & s_{a+2} \\ s_2 & s_3 & s_4 & \ldots & s_{a+2} & s_{a+3} \\ \vdots & \vdots & \vdots & & & \vdots \\ s_b & s_{b+1} & s_{b+2} & \ldots & s_{a+b} & s_{a+b+1} \end{array}\right) \tag{5.3}$$

a non-zero $(b+1) \times (a+2)$ matrix, viewed as system of equations. If $b \ge a$ and the system is consistent, of rank l, then the top l rows of the system are independent, and, in fact, the upper lefthand corner $l \times l$ submatrix,

$$\begin{pmatrix} s_0 & s_1 & s_2 & \ldots & s_{l-1} \\ s_1 & s_2 & s_3 & \ldots & s_l \\ s_2 & s_3 & s_4 & \ldots & s_{l+1} \\ \vdots & \vdots & \vdots & & \\ s_{l-1} & s_l & s_{l+1} & \ldots & s_{2l-2} \end{pmatrix},$$

is invertible.

Proof. Let $t_{a+1}, \ldots, t_1$ be the variables for the equation, in that order. We define the polynomials $s(z) = s_0 + s_1 z + s_2 z^2 + \cdots + s_{a+b+1} z^{a+b+1}$ and $t(z) = 1 + t_1 z + t_2 z^2 + \cdots + t_{a+1} z^{a+1}$ in $F[z]$. The system (5.3) is equivalent to saying that in $s(z)t(z)$ the coefficients of $z^{a+1}, \ldots, z^{a+b+1}$ are 0, or equivalently, that we have

$$s(z)t(z) \equiv u(z) \text{ in } F[z] \text{ modulo } z^{a+b+2} \text{ for some } u(z) \text{ with } \deg(u(z)) \le a\,. \tag{5.4}$$

We are assuming there is a solution to the system, so we have some $t(z)$ and $u(z)$ in $F[z]$ with $s(z)t(z) \equiv u(z)$. If $t(z)$ and $u(z)$ have a common factor $r(z)$, then $r(z)$ has $r(0) \ne 0$ because $t(0) \ne 0$. Replacing $r(z)$ with $r(0)^{-1} r(z)$, we may assume $r(0) = 1$. The class of $r(z)$ is a unit in $F[z]$ modulo z^{a+b+2}: if we write $r(z) = 1 - z\tilde{r}(z)$ for some $\tilde{r}(z)$, we have

$$r(z)(1 + z\tilde{r}(z) + z^2\tilde{r}(z)^2 + \cdots + z^{a+b+1}\tilde{r}(z)^{a+b+1}) = 1 - z^{a+b+2}\tilde{r}(z)^{a+b+2} \equiv 1\,.$$

Multiplying $s(z)t(z) \equiv u(z)$ with this inverse then cancels this common factor in $t(z)$ and $u(z)$. So there is a solution $t(z)$ with matching $u(z)$ of (5.4) for which $t(0) = 1$ and $u(z)$ has no factor in common with $t(z)$.

Let us fix these $t(z)$ and $u(z)$. We can then find another solution of (5.4) by multiplying $t(z)$ and $u(z)$ by any polynomial $r(z)$ with $r(0) = 1$ provided the degree is not too large: we need $\deg(r(z)) \leq c$ with $c = \max(a - \deg(u(z)), a + 1 - \deg(t(z)))$. This, in fact, gives all solutions to (5.4). In order to see this, let $\tilde{t}(z)$ and matching $\tilde{u}(z)$ be another solution to (5.4) with $\deg(\tilde{t}(z)) \leq a + 1$, $\tilde{t}(0) = 1$, and $\deg(\tilde{u}(z)) \leq a$. Then

$$\tilde{t}(z)u(z) \equiv \tilde{t}(z)t(z)s(z) = t(z)\tilde{t}(z)s(z) \equiv t(z)\tilde{u}(z)$$

modulo z^{a+b+2}. But the two polynomials $\tilde{t}(z)u(z)$ and $t(z)\tilde{u}(z)$ have degree at most $2a + 1$, which is less than $a + b + 2$ because $a \leq b$, so that they must be the same in $F[z]$. This means that in the fractional polynomials $F(z)$ we have $\frac{\tilde{u}(z)}{\tilde{t}(z)} = \frac{u(z)}{t(z)}$. The latter is in lowest terms because $u(z)$ and $t(z)$ have no common factor, so $\tilde{t}(z) = r(z)t(z)$ and $\tilde{u}(z) = r(z)u(z)$ for some $r(z)$ as above, because of the degrees of $\tilde{r}(z)$ and $\tilde{u}(z)$.

For $r(z) = 1 + r_1 z + r_2 z^s + \cdots + r_c z^c$ in $F[z]$, the corresponding solution of the system (5.3) is given by the coefficients of $z^{a+1}, z^a, \ldots, z$ in $r(z)t(z)$, in that order. The nullspace of

$$\begin{pmatrix} s_0 & s_1 & s_2 & \cdots & s_a \\ s_1 & s_2 & s_5 & \cdots & s_{a+1} \\ s_2 & s_3 & s_4 & \cdots & s_{a+2} \\ \vdots & \vdots & \vdots & & \\ s_b & s_{b+1} & s_{b+2} & \cdots & s_{a+b} \end{pmatrix} \tag{5.5}$$

is then given by all possible differences of such solutions, which correspond to the coefficients in all possible $t(z)(r_1 z + \cdots + r_c z^c)$. Using that $t(0) = 1$, and letting $r(z) = z, z^2, \ldots, z^c$, we see that the nullspace has a basis consisting of vectors $(\ldots, 1, 0, \ldots, 0)^t$, where the number of 0s at the end is $0, 1, \ldots, c-1$. This means that the last column of (5.5) depends on the first a, the ath column depends on the first $a - 1$, etc., so that the last c columns depend on the first $a + 1 - c$. Because the nullspace has dimension c, the first $a + 1 - c$ columns must be independent, so this number equals the rank l. Also, the arguments until now only used that $b \geq a$, and they remain the same if $b = a$. By symmetry, then also the first l rows of (5.5) are linearly independent, and the next c depend on those. As we are considering only the top $a + 1$ rows of (5.5) at the moment, this shows that the upper left hand corner $l \times l$ submatrix is invertible. As the row rank of the full matrix is also l, it follows that the $b - a$ lowest rows of (5.5) are also dependent on the first l rows. □

Remark 5.6. If we take $1 \leq b < a$, with $s_i = 1$ if $i = a$ and $s_i = 0$ if $i \neq a$, then the resulting system (5.4) is consistent, of rank $b+1$, but the $(b+1) \times (b+1)$ submatrix in the upper lefthand corner is not invertible because its first column consists of zeroes. For example, if we take $a = 2$ and $b = 1$ then this system is

$$\left(\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right).$$

So the condition that $b \geq a$ is necessary.

On the other hand, if we take $b \geq a$, $s_i = 1$ if $i = a + 1$ and $s_i = 0$ if $i \neq a + 1$, then the resulting system (5.4) is inconsistent, the rank of the corresponding matrix (5.5) is a or $a + 1$, but the corresponding $a \times a$ or $(a + 1) \times (a + 1)$ submatrix in the upper lefthand corner is not invertible because its first column consists of zeroes. For example, if we take $a = b = 2$ then this system is

$$\left(\begin{array}{ccc|c} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right)$$

with the corresponding matrix (5.5) of rank 2, and for $a = 2$ and $b = 3$, the system is

$$\left(\begin{array}{ccc|c} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right)$$

with the corresponding matrix (5.5) of rank 3. So the condition that the system (5.4) is consistent is also necessary.