Locally Recoverable Codes

Construction and Properties of Locally Recoverable Codes
Master Thesis Discrete Mathematics

Aalborg University
tenth Semester v/ Faculty of Engineering and Science
Skjernvej 4a • 9220 Aalborg

Figure 1: Locally recoverable cod.

Institute of Mathematical Sciences
Mathematics
Skjernvej 4a
9220 Aalborg Øst
http://math.aau.dk

Title:

Locally Recoverable Codes

Project:

Master Thesis

Project period:

01/02-2021 - 2/06 - 2021

Project group:

Group Mat DISK

Authors:

Rasmus Bod Olesen

Supervisor:

Oliver Wilhelm Gnilke, Matteo Bonini

Pages: 44 + Formalities and appendix.
Finishing date: 02/06-2021
Front page image: Taken from https://andreacullenhealthsolutions.com

Abstract:

This paper analyzes and constructs Locally recoverable codes. The paper initially defines Reed Solomon codes and redundant residue codes and shows the limitations of this particular encoding. The paper then defines locality and constructs Reed Solomon-like locally recoverable codes. Related to this construction is the concept of a nice polynomial and the paper provides several ways of constructing nice polynomials. The paper shows several extensions of the initial construction of locally recoverable codes. Furthermore, the paper also describes how locally recoverable can be constructed by combining several Reed Solomon codes. The paper also proves a singleton-like bound on the minimum distance and shows that almost all the locally recoverable codes constructed in this paper meet this bound with equality. Lastly, the paper defines cyclic locally recoverable codes and describes their subfield subcodes, and shows several results concerning the size of recovering sets for subfield subcodes.

Preface

I developed this master thesis in the period 01-02-2023 to 02-06-2023. It is written by me, Rasmus Bod Olesen. I have had the pleasure of working with my two supervisors Matteo Bonini and Oliver Wilhelm Gnilke and I would like to thank both of them for their invaluable assistance and insight.

Since this is (hopefully) my last project I would also like to extend my thanks to my family who have been unwavering in their support they are my recovering set.

Signature

Rasmus Bod Olesen

Contents

List of Symbols

- $\mathbb{F}_q$. A finite field with $q = p^m$ elements for some prime p and integer m
- $\mathbb{F}$. A finite field with arbitrarily many elements.
- x_i. The i'th coordinate in the vector x
- C_I the code C with the entries not in I removed. So if $I = \{1\}$ we remove entry 1 in all codewords of C
- $\sqcup$. The disjoint union which retains set membership. So $A \sqcup B = (A \times 0) \cup (B \times 1)$

1 Introduction

In recent years distributed storage systems have become increasingly common. On such data storage systems, people often have to access the same data multiple times we refer to such data as "hot data" and when the demand for the data exceeds server capacity a temporary failure can occur which causes the "hot data" to become inaccessible (Li & Li 2013). To facilitate access to hot data specific encoding of the data is required. One of the more common encoding methods for data recovery is replication where several copies of the data fragment are stored at different storage nodes. Note here that by a node, we mean a storage unit. However, this type of encoding is insufficient in many cases since a large amount of overhead is required. Therefore different encoding techniques are utilized to achieve lower overhead with similar or comparable resistance to failure. For example, to minimize storage overhead Reed Solomon codes (RS-codes) are often utilized (Tamo & Barg 2014).

When a file is encoded using RS-codes the file is first partitioned into k fragments which are then encoded so $n - k$ fragments are added as redundancy and the n data fragments are stored over n nodes. This type of encoding is already implemented in several storage systems. For example, Hadoop Distributed File System uses a $(9, 6)$ Reed Solomon code which means that the storage overhead is $\frac{n}{k} = \frac{9}{6} = \frac{3}{2}$ (Ramkumar et al. 2022).

Although RS-codes are great one of the main issues in terms of node failure is that in order to recover a single node we must read k other nodes which is time-consuming (Tamo 2015) (Tamo & Barg 2014). Therefore, locally recoverable codes were constructed to fix this issue. A locally recoverable code is a code in which each symbol of the codeword can be recovered from r other symbols where $r \leq k$. The downside to utilizing these codes is that the code constructed no longer meets the MDS bound with equality. However, a new MDS-like bound is achieved and most of the LRC codes constructed in this paper meet this bound with equality (Tamo & Barg 2014) Ramkumar et al. (2022).

2 Reed Solomon Codes and Redundant Residue Codes

This chapter is based on (MacWilliams 1983) and (Tamo 2015). We initially discuss Reed Solomon codes as the initial construction of locally recoverable codes (shortened LRC codes) is an extension of Reed Solomon codes. The general definition is given in Definition 2.0.1.

Definition 2.0.1
Given a message $m \in \mathbb{F}^k$ define the encoding polynomial

$$f_m(x) = \sum_{i=0}^{k-1} m_i x^i. \tag{2.1}$$

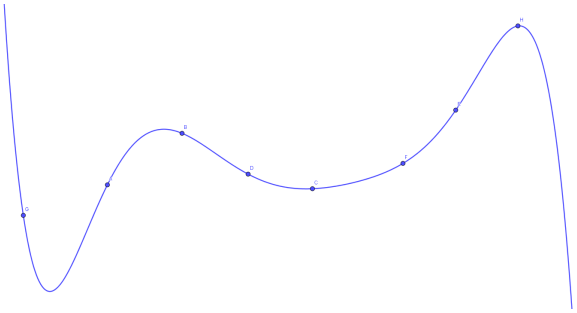
For a given evaluation vector $\alpha = (\alpha_1, ..., \alpha_n) \in \mathbb{F}^n$ we define the RS encoding function as $RS_{n,k}(\alpha, m) = (f_m(\alpha_1), ..., f_m(\alpha_n))$. The set $\{RS_{n,k}(\alpha, m), m \in \mathbb{F}^k\}$ is then defined as the Reed Solomon code.

One of the reasons the Reed Solomon codes are interesting is that the code obtains equality in the singleton bound which is stated here without proof.

Theorem 2.0.2
For a linear (n, k) code with minimum distance d_m the following bound holds.

$$d_m \leq n - k + 1.$$

As mentioned the RS code obtains equality in this bound. Codes that meet the singleton bound are called *Maximum Distance separable* codes shortened MDS codes.
We illustrate the problems with RS-codes in Figure 2.1.



(a)

Figure 2.1: The figure describes the Reed Solomon encoding. We have a polynomial of degree $k-1$ evaluated at 8 points. If a symbol is lost it can be recovered by performing polynomial interpolation through the remaining points. To perform the interpolation we must access at least k points. This is the problem with the RS-codes in order to recover we must access k symbols which when k is large or when several recoveries are necessary is not feasible.

So the problem with reed Solomon codes is that you must read k symbols. So the future constructions of locally recoverable codes will utilize a similar recovery procedure based on polynomial interpolation but will read fewer symbols.

3 Locally Recoverable Codes Definition and Properties

The following chapter is based on (Tamo & Barg 2014) and Lauritzen (2003). Initially, the definition of *locality* is given. A code C is said to have locality r if every symbol of the codeword $x \in C$ can be recovered from a subset of r other symbols. More formally locality is given in Definition 3.0.1.

Definition 3.0.1
Consider $a \in \mathbb{F}_q$ and define a set of codewords as

$$\mathcal{C}(i,a) = \{x \in \mathcal{C} : x_i = a\}, \quad i \in 1, ..., n.$$

The code $\mathcal{C}$ has locality r if for every $i \in \{1, ..., n\}$ there exists a subset $I_i \subset \{1, ..., n\} \backslash i$ where $|I_i| \leq r$ such that

$$C_{I_i}(i,a) \cap C_{I_i}(i,a') = \emptyset, \quad a \neq a'.$$

A code with the locality property is called a locally recoverable code. For each codeword, we have a unique recovering set I_i of size at most r; from this set, the symbol of the codeword can be recovered. We also note that the locality parameter must satisfy $1 \leq r \leq k$ because if we read k symbols we can always recover the codeword. For these codes, we can establish the following minimum distance and rate.

Theorem 3.0.2
Let $\mathcal{C}$ be an (n, k, r) LRC code then:

$$\frac{k}{n} \leq \frac{r}{r+1} \tag{3.1}$$

$$d \leq n - k - \lceil \frac{k}{r} \rceil + 2. \tag{3.2}$$

This theorem is proved in Chapter 6. We now construct a linear code with the locality property. This construction relies on a polynomial g with the following properties:

- $\deg(g) = r + 1$

- There exists a partition of the set $A \subseteq \mathbb{F}_q$ denoted $\mathcal{A} = \{A_1, A_2, ..., A_{\frac{n}{r+1}}\}$ such that for all $i \in 1, ..., \frac{n}{r+1}$ and any $\alpha, \beta \in A_i$ we have that

$$g(\alpha) = g(\beta).$$

So g is constant on each set A_i. A polynomial with this property is called *nice*. Here it is assumed that $r + 1|n$. We will later show that such a polynomial and such a partition can almost always be found. We now state the construction of a linear locally recoverable code.

Construction 3.0.3
Let $n \leq q$ be the target code length and let $A \subset \mathbb{F}_q$ where $|A| = n$ and let $g(x)$ be a nice polynomial for the partition $\mathcal{A}$. If we wish to encode the message $m \in \mathbb{F}_q^k$ we write it as $m = (m_{ij}, i = 0, ..., r-1; j = 0, ..., \frac{k}{r} - 1)$. We then define the encoding polynomial as

$$f_m(x) = \sum_{i=0}^{r-1} f_i(x)x^i, \tag{3.3}$$

where

$$f_i(x) = \sum_{j=0}^{\frac{k}{r}-1} m_{ij}g(x)^j \tag{3.4}$$

The codewords from this encoding polynomial is then defined as

$$C = \{(f_m(\alpha), \alpha \in A) : m \in \mathbb{F}_q^k\}$$

Where $(f_m(\alpha), \alpha \in A)$ is an evaluation vector defined from the set A.

Note here that this construction relies on $k|r$ however this assumption can be neglected as Proposition 3.0.4 suggests.

Proposition 3.0.4
The assumption $r|k$ is not necessary.

Proof. Assume that $r \nmid k$ and let $k = d \cdot r + R$. We start by indexing our message vector

in a new way which is represented in the following vectors:

$$\begin{aligned} &m_{i,j} = \\ &(m_{0,0}, ..., m_{0,\lfloor \frac{k}{r} \rfloor}) \\ &\vdots \\ &(m_{R-1,0}, ..., m_{R-1,\lfloor \frac{k}{r} \rfloor}) \\ &(m_{R,0}, ..., m_{R,\lfloor \frac{k}{r} \rfloor - 1}) \\ &\vdots \\ &(m_{r-1,0}, ..., m_{r-1,\lfloor \frac{k}{r} \rfloor - 1}). \end{aligned}$$

By indexing, this way it is clear that all k symbols are preserved as the number of message symbols in this new set of vectors is k. We now define our coefficient polynomial in the following way utilizing our new indexing.

$$f_i(x) = \sum_{j=0}^{s(k,r,i)} m_{ij} g(x)^j, \quad i = 0, 1, ..., r-1.$$

Here $s(k, r, i)$ is defined by the following function.

$$s(k, r, i) = \begin{cases} \lfloor \frac{k}{r} \rfloor & \text{if } i < R \\ \lfloor \frac{k}{r} \rfloor - 1 & \text{if } i \geq R \end{cases}.$$

Using this function we still encode all k elements from our message vector m. □

The notation for this is cumbersome so for future proofs we just assume $r|k$. We can also state Proposition 3.0.5.

Proposition 3.0.5
The code constructed in Construction 3.0.3 is linear.

Proof. We show that the encoding polynomial is linear for a given evaluation α as this implies that the set of codewords is linear. So let $p \in \mathbb{F}_q, m, m' \in \mathbb{F}_q^k$ then consider the

following:

$$\begin{aligned}
&(f_m(\alpha)) + (f_{m'}(\alpha)) \\
&= \sum_{i=0}^{r-1} f_i(\alpha)\alpha^i + f_i(\alpha)\alpha^i \\
&= \sum_{i=0}^{r-1} \alpha^i \sum_{j=0}^{k/r-1} m_{ij}g(\alpha)^j + m'_{ij}g(\alpha)^j \\
&= \sum_{i=0}^{r-1} \alpha^i \sum_{j=0}^{k/r-1} g(\alpha)^j(m_{ij} + m'_{ij}) \\
&= f_{m+m'}(\alpha) \in C.
\end{aligned}$$

Now consider the following:

$$\begin{aligned}
&p \cdot f_m(\alpha) \\
&= \sum_{i=0}^{r-1} pf_i(\alpha)\alpha^i \\
&= \sum_{i=0}^{r-1} \alpha^i \sum_{j=0}^{k/r-1} g(\alpha)^j \cdot (p \cdot m_{ij}) \\
&= f_{p\cdot m}(\alpha) \in C.
\end{aligned}$$

□

We also require a way to decode the codewords in Construction 3.0.3 to do this consider a codeword $c_\alpha \in C$. An erasure occurs at a symbol corresponding to $\alpha \in A_j$ now let $(c_\beta, \beta \in A_j \backslash \alpha)$ denote the symbols in the set A_j of size r. We must find the value $c_\alpha = f_m(\alpha)$ therefore we construct the unique polynomial $\delta(x)$ given as:

$$\delta(x) = \sum_{\beta \in A_j \backslash \alpha} c_\beta \prod_{\beta' \in A_j \backslash \{\alpha,\beta\}} \frac{x - \beta'}{\beta - \beta'}. \tag{3.5}$$

Let $c_\alpha = \delta(\alpha)$. Note here that $\delta(\beta) = c_\beta$. We will now state the following theorem concerning the recovery procedure and the locality.

Theorem 3.0.6
The Code C constructed in 3.0.3 has dimension k, obtains equality in Equation (3.2) and has locality r.

Proof. First, note that all the polynomials $g(x)^j x^i$ are all of distinct degrees for all $i = 0, ..., r-1; j = 0, ..., \frac{k}{r} - 1$. Therefore polynomials $g(x)^j x^i$ are linearly independent. This

then implies that the mapping $m \to f_m(x)$ is injective. Also the degree of $f_m(x)$ can be determined to be at most

$$(\frac{k}{r} - 1)(r+1) + r - 1 = k + \frac{k}{r} - 2.$$

This follows from the fact that $\deg(g(x)^j) = (\frac{k}{r} - 1)(r+1)$ for $j = \frac{k}{r} - 1$. Since the mapping is injective two distinct polynomials $f_m, f_{m'}$ give two distinct codewords. Therefore the dimension of the code must be k. Also, the encoding is linear therefore the distance of the code must satisfy

$$\begin{aligned} &d_H((f_m(\alpha), \alpha \in A), (f_{m'}(\alpha), \alpha \in A)) \\ &= n - \mathrm{wt}((f_m(\alpha) - f_{m'}(\alpha)), \alpha \in A) \\ &= n - \mathrm{wt}((f_{m-m'}(\alpha), \alpha \in A)) \\ &\geq n - (k + \frac{k}{r} - 2) = n - k - \frac{k}{r} + 2. \end{aligned}$$

Because the mapping from $\mathbb{F}^k$ to $\mathbb{F}^n$ is injective we get that the weight is non-zero and since this holds for the hamming distance it must hold for the minimum distance and so we get.

$$d_m \geq n - \max_{f_m(\alpha), m \in \mathbb{F}_q^k} \deg(f_m) = n - k - \frac{k}{r} + 2. \tag{3.6}$$

Together with Equation (3.2) it is now clear that the code constructed obtains equality in the desired bound. We must also prove the code satisfies the locality property. Assume that the symbol $c_\alpha = f_m(\alpha)$ has been lost where $\alpha \in A_j \subset F_q$. We define a new polynomial $\partial(x)$ and show that it is equal to the polynomial given in Equation (3.5).

$$\partial(x) = \sum_{i=0}^{r-1} f_i(\alpha) x^i.$$

Since $g(x)$ is constant on the sets A_j so are the $f_i(x)$ this implies that

$$\partial(\beta) = \sum_{i=0}^{r-1} f_i(\alpha)\beta^i = \sum_{i=0}^{r-1} f_i(\beta)\beta^i = f_m(\beta).$$

Note that $\deg(\partial(x)) \leq r - 1$ so the polynomial can be interpolated from the r symbols in $c_\beta, \beta \in A_j \backslash \alpha$. Therefore it is equal to the polynomial $\delta(x)$ since they were interpolated from the same symbols. So the lost symbol c_α can be recovered by using r other symbols in the code word which is the definition of locality. □

We now compare this construction with the Reed Solomon codes in Figure 3.1.

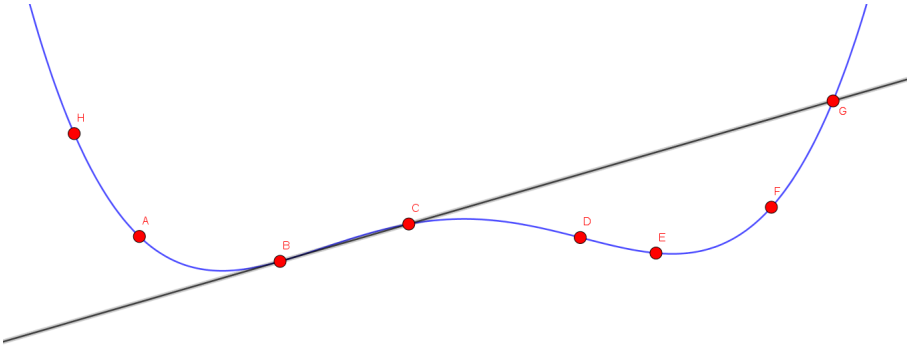


Figure 3.1: The encoding polynomial in blue. In RS codes if the coordinate C is lost we would need to perform interpolation from 4 other coordinates. However, because the code has locality 2 we know that a polynomial of degree 1 passes through our point and so the coordinate C can be recovered from the points B, G. So for all points, there exists a polynomial of degree 1 from which the point can be recovered.

The main issue with this construction is finding a polynomial g which is constant on A_i. To do this more systematically we shall rely on the multiplicative subgroups and additive subgroups of $\mathbb{F}_q$. We will show that such a polynomial and such a set can be formed by Proposition 3.0.7.

Proposition 3.0.7
Let H be a subgroup of $\mathbb{F}_q^\cdot$ or $\mathbb{F}_q^+$ and. The polynomial

$$g(x) = \prod_{h \in H} (x - h)$$

is constant on each coset of H.

Proof. Let $a, a\bar{h}$ be two elements of the coset aH then:

$$g(a\bar{h}) = \prod_{h \in H} (a\bar{h} - h) = \bar{h}^{|H|} \prod_{h \in H} (a - h\bar{h}^{-1}) = \prod_{h \in H} (a - h) = g(a).$$

□

For a multiplicative group, this can be further reduced. For any group G we have that $g^{|G|} = 1$ for $g \in G$. Therefore every element of the multiplicative subgroup H is a root in the polynomial $p(x) = x^{|H|} - 1$ and since the polynomial can have at most $|H|$ roots and since they are both monic we get the following:

$$g(x) = \prod_{h \in H} (x - h) = x^{|H|} - 1. \tag{3.7}$$

Example 3.0.8.
In this example, we utilize Proposition 3.0.7 to construct a (14,12,6) LRC code over the field $\mathbb{F}_{29}$. Note first that 7 is a $r+1 = 6+1 = 7$th root of unity and consider the multiplicative subgroup generated by 7 $\langle 7 \rangle = \{1, 7, 20, 24, 23, 16, 25\}$ and the coset $2 \cdot \langle 7 \rangle = \{2, 14, 11, 19, 17, 3, 21\}$. These two sets make up our partition $\mathcal{A}$.

$$\mathcal{A} = \{\{1, 7, 20, 24, 23, 16, 25\}, \{2, 14, 11, 19, 17, 3, 21\}\}.$$

Our nice polynomial is $x^7 - 1$ as given in Equation (3.7). The encoding polynomial will be as follows

$$f_m(x) = \sum_{i=0}^{5} f_i(x)x^i = \sum_{i=0}^{5} (m_{i,0} + m_{i,1}(x^7 - 1))x^i.$$

The codeword is generated by evaluating $f_m(x)$ for all $x \in A$.

As the next example shows, we can also construct a field based on the additive subgroup.

Example 3.0.9.
In this example, we generate an LRC code (14,12,6) code over the field $\mathbb{F}_{16}$. Let α be a primitive element of the field $\mathbb{F}_{16}$ so every element of $\mathbb{F}_{16}$ can be written as α^i for some $i \in \mathbb{N}$. We then form the additive subgroup $H = \{x + y\alpha : x, y \in \mathbb{F}_2\}$. We can calculate the polynomial g as

$$g(x) = x(x+1)(x+\alpha)(x+\alpha+1).$$

Similarly to Example 3.0.8 we can form the encoding polynomials $f_m(x)$.

Combining these two methods results in a more general construction method of nice polynomials. To construct such polynomials we need Definition 3.0.10.

Definition 3.0.10
For two subsets $H, G \subset F_q$ we say that H is closed under multiplication by G if

$$\{hg : h \in H, g \in G\} \subseteq H.$$

So multiplying elements of G with elements from H results in an element still contained in H. Using this definition we can state Theorem 3.0.11.

Theorem 3.0.11
Let $l, s, w \in \mathbb{N}$ be given such that:

- $l | s$.
- $p^l \bmod w = 1$, where p is prime.

Let H be an additive subgroup of $\mathbb{F}_{p^s}$ closed under multiplication by $\mathbb{F}_{p^l}$ and let $\alpha_1, ..., \alpha_w$ be the w w-th degree roots of unity in $\mathbb{F}_{p^l}$ which we know exist since $p^l \bmod w = 1$. Then

for any $b \in \mathbb{F}_{p^s}$ the polynomial:

$$g(x) = \prod_{i=1}^{w} \prod_{h \in H} (x + h + \alpha_i) \tag{3.8}$$

is constant on the union of cosets $\cup_{1 \leq i \leq w} H + b\alpha_i$. Furthermore, the cardinality of this union is.

$$|\cup_{1 \leq i \leq w} H + b\alpha_i| = \begin{cases} |H| & \text{if } b \in H \\ w|H| & \text{if } b \notin H. \end{cases}$$

Proof. Let $\bar{h} \in H$ and let $\bar{h} + b\alpha_j \in H + b\alpha_j$. We can then do the following calculation:

$$\begin{aligned} g(\bar{h} + b\alpha_j) &= \prod_{i=1}^{w} \prod_{h \in H} (\bar{h} + b\alpha_j + h + \alpha_i) \\ &= \prod_{i=1}^{w} \prod_{h \in H} (b\alpha_j + h + \alpha_i) \\ &= \alpha_j^{-w|H|} \prod_{i=1}^{w} \prod_{h \in H} (b + h\alpha_j^{-1} + \alpha_i \alpha_j^{-1}) \\ &= \prod_{i=1}^{w} \prod_{h \in H} (b + h\alpha_j^{-1} + \alpha_i) \\ &= \prod_{i=1}^{w} \prod_{h \in H} (b + h + \alpha_i) \\ &= g(b). \end{aligned}$$

Where the second-to-last equality holds because H is closed under multiplication by elements of $\mathbb{F}_{p^l}$, the size of the union follows from the series of equivalences. Take two distinct roots of unity α_i, α_j then

$$H + b\alpha_i = H + b\alpha_j \iff b(\alpha_i - \alpha_j) \in H \iff b \in H$$

Here the last iff statement holds because $\alpha_i - \alpha_j$ is an element of $\mathbb{F}_{p^l}$ and H is by definition closed under multiplication by elements in $\mathbb{F}_{p^l}$. This implies that if $b \in H$ then all the cosets are equal and thus $|\cup_{1 \leq i \leq w} H + b\alpha_i| = |H|$. If $b \notin H$ then $H + b\alpha_i \neq H + b\alpha_j$. By group theory, this implies that $H + b\alpha_i \cap H + b\alpha_j = \emptyset$ See Lemma 2.2.6 in Lauritzen (2003). Since the intersection is empty we get that $|\cup_{1 \leq i \leq w} H + b\alpha_i| = w|H|$. □

In order to use Theorem 3.0.11 we need to find a subgroup H of $\mathbb{F}_{p^s}$ which is closed under multiplication by $\mathbb{F}_{p^l}$. So we utilize the fact that $\mathbb{F}_{p_s}$ can be viewed as a vector space over the field F_{p^l} of dimension $\frac{s}{l}$. So if we pick a subspace H of the vector space $\mathbb{F}_{p^s}$ then H is

an additive subgroup which is closed under multiplication by $\mathbb{F}_{p^l}$ where $\dim(H) = t \leq \frac{s}{l}$ and $|H| = (p^l)^t = p^{tl}$.
Furthermore, the polynomial $g(x)$ takes distinct values on all sets U where $U = \cup_{1\leq i\leq w} H + b\alpha_i$ if it did not we would have that the polynomial $g(x) - c$ would have more than $w|H|$ roots for some $c \in \mathbb{F}_{p^s}$. This is not possible as $\deg(g(x)) = w|H|$. This means that from $g(x)$ we can partition $\mathbb{F}_{p^s}$ into $\frac{p^s-|H|}{w|H|}$ sets of size $w|H|$ and one set of size $|H|$. Again we do an example to illustrate how to find a nice polynomial using Theorem 3.0.11.

Example 3.0.12.
Consider $p = 7$ and say we wish to construct a $(30, 14) = (n, r)$ LRC code. We set $m = 3, l = 1, s = 3$ the three 3rd roots of unity over $\mathbb{F}_7$ are $1, 2, 4$. If $t = 1$ we can set $H = \mathbb{F}_7^+$ as the additive subgroup since it is closed under multiplication by the field $\mathbb{F}_7$. So we can construct our nice polynomial g using Equation (3.8).

$$g(x) = \prod_{i=1}^{3} \prod_{h\in H} (x + h + \alpha_i).$$

This polynomial will then split the field $\mathbb{F}_{7^3}$ into 16 sets of size 21 and one set of size 7.

We can summarize the ways of constructing nice polynomials suppose we have a finite field $\mathbb{F}_{p^l}$ and wish to construct a nice polynomial that is constant on disjoint sets of size wp^t where $\gcd(w, p) = 1$.

- If $t = 0$ we can use the multiplicative subgroup can be used as is done in Example 3.0.8. This can be done if $p^l \bmod w = 1$.
- If $t > 0$ and $w = 1$ the additive subgroup can be used as is done in Example 3.0.9.
- If $t, w > 1$ and t is a multiple of l one can use Theorem 3.0.11. Provided that l is the smallest integer such that $p^l \bmod w = 1$. This is done in Example 3.0.12

There are a few cases where we are not able to construct LRC codes. Consider the LRC code over the field $\mathbb{F}_{2^l}$ with locality $r = 5$. This can not be done as the size of the set must be $r + 1 = 6 = 2 \cdot 3$ so $w = 3$ and so the smallest solution l to the equation $2^l \bmod 3 = 1$ is $l = 2$ but if we pick $t = 1$ this is not a multiple of 2 and so no such code can exist. We now state a proposition related to the existence of nice polynomials.

Proposition 3.0.13
Let $\mathbb{F}_q$ be a finite field. There exists a nice polynomial g where $\deg(g) = r + 1$ which is constant on at least $\lceil \frac{\binom{q}{r+1}}{q^r} \rceil$ sets of size $r + 1$.

Proof. We first define the set

$$M_{q,r} = \{f \in \mathbb{F}_q[x] | f = \prod_{i=1}^{r+1} (x - \alpha_i)\},$$

where α_i varies over the $\binom{q}{r+1}$ choices of subsets of size $r+1$. This means that $M_{q,r}$ becomes the set of monic polynomials where $\deg(f) = r+1$ for $f \in M_{q,r}$ and where f has $r+1$ distinct roots in $\mathbb{F}_q$. We now define an equivalence relation on $M_{q,r}$ for two polynomials $f(x) = x^{r+1} + \sum_{i=0}^{r} a_i x^i$ and $g(x) \in M_{q,r}$ we say that $f(x) \sim g(x)$ if they differ by a constant. $\sim$ defines a equivalence relation over $M_{q,r}$.
The number of equivalence classes is at most q^r because we have q choices for each a_i. This implies that there is an equivalence class of size at least $\lceil \frac{\binom{q}{r+1}}{q^r} \rceil$ because there are $\binom{q}{r}$ choices for a polynomial from $M_{q,r}$. Now we let f, g be two elements of this equivalence class by the transitive property we have that $f\ g$ so f, g differ by a constant and so f is constant on the set of roots of g. This implies that f is a nice polynomial that is constant on sets of size $r+1$ and the amount of sets is at least $\lceil \frac{\binom{q}{r+1}}{q^r} \rceil$. □

Proposition 3.0.13 gives us certainty that nice polynomials exist provided that q is large enough.

4 Generalization

In this chapter, we extend LRC codes and construct a family of codes that are not necessarily linear to do this we utilize a map from the set of polynomials given in Equation (3.3) to $\mathbb{F}^n$. To do this we first define the following set of polynomials.

$$\mathbb{F}_{\mathcal{A}}[x] = \{f \in \mathbb{F}[x] | f \text{ is constant on } A_i, i, ..., w; \deg(f) < |A|\}.$$

So this is the set of polynomials with $\deg(f) < |A|$ such that f is constant on each of the sets A_i. This set of polynomials can form a commutative algebra if calculated modulo a polynomial $h(x)$. The polynomial $h(x)$ is defined as follows.

$$h(x) = \prod_{a \in A} (x - a).$$

So $h(a) = 0$ if $a \in A$. So in the following section, we will take fg to mean $fg \bmod h$. We can now state Proposition 4.0.1 which concerns the properties of the algebra.

Proposition 4.0.1

- If $f \in \mathbb{F}_{\mathcal{A}}[x]$ is non-constant then $\max_i |A_i| \leq \deg(f)$.
- We have that $\dim(\mathbb{F}_{\mathcal{A}}[x]) = w$. Furthermore, the w polynomials $f_i, i = 1, ..., w$ with the following properties: $f_i(A_j) = \partial_{i,j}$ where $\partial_{i,j} = 1$ if $i = j$ and $\partial_{i,j} = 0$ if $i \neq j$ and $\deg(f_i) < |A|$. Form a basis if $x \in A$ and the polynomials of this basis can be written explicitly as:
$$f_i(x) = \sum_{a \in A_i} \prod_{a \in A \setminus a} \frac{x - b}{a - b}.$$
- Let $\alpha_i, ..., \alpha_w \in \mathbb{F}$ and let $g \in \mathbb{F}[x]$ be the polynomial where $\deg(g) < |H|$ and $g(A_i) = \alpha_i$ for all $i = 0, ..., w$ given explicitly as:
$$g(x) = \sum_{i=1}^{w} \alpha_i \sum_{a \in A_i} \prod_{b \in A \setminus a} \frac{x - b}{a - b}.$$
Then the polynomials $1, g, ..., g^{w-1}$ form a basis of $\mathbb{F}_{\mathcal{A}}[x]$ if $x \in A$.

- There exists w integers $0 = d_0 < d_1 < d... < d_{w-1} < |A|$. Such that for all $f \in \mathbb{F}_{\mathcal{A}}[x]$ and some d_i we have that $\deg(f) = d_i$.

Proof.

- Consider a polynomial $f \in \mathbb{F}_{\mathcal{A}}[x]$ and a set $A_i \in \mathcal{A}$, then the polynomial $f(x) - f(A_i)$ has at least $|A_i|$ roots in $\mathbb{F}$ and so $\deg(f) \geq |A_i|$ and since we can choose A_i such that $|A_j| \leq |A_i|$ for all $A_j \in \mathcal{A}$ it follows that $\max_i |A_i| \leq \deg(f) < |A|$.
- The polynomials are linearly independent if $x \in A$. As the following calculation shows.

$$\sum_{i=1}^{w} \lambda_i f_i(x) = 0$$

We must show that this implies that $\lambda_i = 0$. So for any $j \in \{1, ..., w\}$.

$$\sum_{i=1}^{w} \lambda_i f_i(A_j) = \sum_{i=1}^{w} \lambda_i \partial_{i,j} = \lambda_j = 0.$$

Since the $f_1, ..., f_w$ span all $\mathbb{F}_{\mathcal{A}}[x]$ we have that $\dim(F_{\mathcal{A}}[x]) = w$.

- Let $\beta_j \in \mathbb{F}$ be given and then assume

$$\sum_{j=1}^{w} \beta_j g^{j-1}(x) = 0.$$

We can write this sum as a Vandermonde matrix multiplied by a vector so let $(v_{i,j}) = (g^{j-1}(A_i))$. We can then write the equation above as $V \circ (b_1, ..., b_w)^T = 0$. But since V is a Vandermonde matrix it is invertible because it is defined from m distinct elements α_i and so $\beta_i = 0$ for all i.

- Now let $f_0, ..., f_{w-1}$ be a basis for $\mathbb{F}_{\mathcal{A}}[x]$. We can assume without loss of generality that the polynomials all have distinct degrees. If they did we can simply perform row operations on the matrix whose rows consist of the coefficient vectors from the polynomials $f_0, ..., f_{w-1}$. This $w \times |A|$ matrix would have reduced row echelon form and form a basis with polynomials of distinct degrees $d_i = \deg(f_i)$. Furthermore, the constant polynomials are contained in $\mathbb{F}_{\mathcal{A}}[x]$ this means that $d_0 = 0$ and so this concludes the proof.

□

Before we state the extended construction we first prove Corollary 4.0.2.

Corollary 4.0.2
Assume the partition satisfies $|A_i| = r + 1$ and that there exists a polynomial $\deg(g) =$

$d_1 = r + 1$ then $d_i = i(r + 1)$ for all $i \in 0, ..., w - 1$ and the polynomials $1, g, .., g^{w-1}$ defined in Proposition 4.0.1 form a basis for $\mathbb{F}_{\mathcal{A}}$.

Proof. If the polynomial exists then it takes distinct values on distinct sets of the partition $\mathcal{A}$. Because if it did not the polynomial $g(x) - c$ where $c \in \mathbb{F}$ would have at least $2(r + 1)$ roots which is not possible since $\deg(g) = r + 1$ and so g satisfies the conditions in Proposition 4.0.1 part 3 and so the polynomials form a basis. □

Since $r + 1|n$ we can always find a nice polynomial in $\mathbb{F}_{\mathcal{A}}[x]$ with degree $r + 1$. This is because there exists a unique subgroup of order $r + 1$ and thus $g(x) = x^{r+1} - 1$ is such a polynomial of degree $r + 1$. Now that we have described the properties of the algebra we will now use it to construct a (n, k, r) LRC code.

Construction 4.0.3
Let $A \subset \mathbb{F}$, $|A| = n$ and let $\mathcal{A}$ be a partition of A into $w = \frac{n}{r+1}$ sets of size $r + 1$. Let Φ be an injective map from $\mathbb{F}^k$ to the space of polynomials:

$$\mathcal{F}^r_{\mathcal{A}} = \oplus_{i=0}^{r-1} \mathbb{F}_{\mathcal{A}}[x]x^i.$$

Note here that Φ exists iff $k \leq wr = \frac{nr}{r+1}$. This is because $\dim(\mathcal{F}^r_{\mathcal{A}}) = wr$ because it is a direct sum of spaces.

The function Φ maps messages $m \in \mathbb{F}^k$ to a set of polynomials that encode the message. The code is then given by evaluating the polynomials $f \in \Phi(\mathbb{F}^k)$ at the points of A. If Φ is linear then the code is also linear. We state a proof similar to Theorem 3.0.6 for Construction 4.0.3. Note here that we write $\Phi(m, x)$ to denote the unique polynomial which corresponds to our message m evaluated at x.

Theorem 4.0.4
The code constructed in 4.0.3 is an (n, k, r) LRC code satisfying

$$d \geq n - \max_{m,m' \in \mathbb{F}^k} (\deg(\Phi(m, x) - \Phi(m', x)) \geq n - \max_{m \in \mathbb{F}^k} \deg(\Phi(m, x)).$$

Proof. The proof of locality is similar to the proof of locality in Theorem 3.0.6. But it is stated here for completion. Let $m \in \mathbb{F}^k$ be a given message. We let

$$\Phi(m, x) = \sum_{i=0}^{r-1} f_i(x)x^i. \tag{4.1}$$

This holds because $\mathcal{F}^r_A$ is a direct sum of the spaces $\mathbb{F}_{\mathcal{A}}[x]x^i$ for $i = 0, 1, ..., r - 1$. We now pick $j \in \{1, ..., m\}$ and suppose an erasure occurs at $\Phi(m, \alpha)$. We then define the decoding

polynomial

$$\delta(x) = \sum_{i=0}^{r-1} f_i(\alpha)x^i.$$

We note that $\delta(\alpha) = \Phi(m, \alpha)$ and because $f_i \in \mathbb{F}_{\mathcal{A}}[x]$ we have that for any $\beta \in A_j$ the following holds $\Phi(m, \beta) = \delta(\beta)$. Also $\deg(\delta(x)) \leq r - 1$ which means that $\delta(x)$ can be interpolated by utilizing the r symbols $\Phi(m, \beta) = \delta(\beta)$ where $\beta \in A_j \backslash \alpha$.
For proof of the inequalities consider the following:

$$\begin{aligned}
&d_H(\Phi(m, x), \Phi(m', x)) \\
&= \mathrm{wt}(\Phi(m, x) - \Phi(m', x)) \\
&= n - |\{\alpha | \Phi(m, x) - \Phi(m', x) = 0\}| \\
&\geq n - \deg(\Phi(m, x) - \Phi(m', x)) \\
&\geq n - \max_{m,m' \in \mathbb{F}^k} (\deg(\Phi(m, x) - \Phi(m', x)) \\
&\geq n - \max_{m \in \mathbb{F}^k} (\deg(\Phi(m, x))).
\end{aligned}$$

Since the inequality holds for the hamming distance it must hold for the minimum distance. □

4.1 Systematic Encoding

A code C is said to encode systematically if the input data is embedded in the encoding output. For linear codes, this can be done by utilizing the following generator matrix $G = (I, P)$ where I is a $k \times k$ identity matrix. This is a desirable characteristic as it makes the retrieval of the message easier, therefore we discuss a modification of the LRC codes which ensures systematic encoding.

So let $\mathcal{A} = \{A_1, ..., A_w\}$ where $w = \frac{n}{r+1}$ be a partition of the set A into sets of size $r + 1$. For $i = 1, .., \frac{k}{r}$ define the subset of A_i:

$$B_i = \{\beta_{i,1}, ..., \beta_{i,r}\}$$

and note that the size of the subset is r. By Proposition 4.0.1 we have that $f_i(A_j) = \partial_{i,j}$ for $i, j = 1, ..., w$ forms a basis for the algebra $\mathbb{F}_{\mathcal{A}}[x]$. For each set B_i we define r polynomials $\phi_{i,j}$ where $\deg(\phi_{i,j}) < r$

$$\phi_{i,j}(\beta_{i,l}) = \partial_{j,l}.$$

Note that such polynomials can be found using Lagrange Interpolation. Let the message $m = (m_{i,j}), i = 1, ..., \frac{k}{r}; j = 1, ..., r$ be given. The following encoding polynomial is then defined.

$$f_m(x) = \sum_{i=1}^{\frac{k}{r}} \left(\sum_{j=1}^{r} m_{i,j}\phi_{i,j}(x) \right).$$

Again the codeword is given as the vector $(f_m(\alpha), \alpha \in A)$. The code has locality r as $f_m \in \mathcal{F}^r_{\mathcal{A}}$. Finally, we have that

$$f_m(\beta_{i,j}) = m_{i,j}$$

for $i = 1, ..., \frac{k}{r}$ and $j = 1, ..., r$. So this implies that the code is systematic. Even though the encoding is systematic there is a problem with the encoding, as the minimum distance is not optimal in terms of Equation (3.2). This is because we can only guarantee that $\deg(f_m) < n$. However, this problem can be alleviated by utilizing a nice polynomial g whose powers generate $\mathbb{F}_{\mathcal{A}}[x]$ this is possible because of Proposition 4.0.1 part 3. So we can replace the polynomials in f_i with polynomials $\bar{f_i}$ which are a linear combination of the nice polynomials $1, g, .., g^{\frac{k}{r}-1}$ which satisfy $\bar{f_i}(A_j) = \partial_{i,j}$. Note that this is possible because the matrix $V = (g^{j-1}(A_i))$ is a $\frac{k}{r} - 1 \times \frac{k}{r} - 1$ Vandermonde matrix and is therefore invertible. To see this consider the solution to the equation

$$\begin{bmatrix} 1 & g(A_1) & g^2(A_1) & \dots & g^{\frac{k}{r}-1}(A_1) \\ 1 & g(A_2) & g^2(A_2) & \dots & g^{\frac{k}{r}-1}(A_2) \\ & & \vdots & & \\ 1 & g(A_{\frac{k}{r}}) & g^2(A_{\frac{k}{r}}) & \dots & g^{\frac{k}{r}-1}(A_{\frac{k}{r}}) \end{bmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{\frac{k}{r}} \end{pmatrix} = e_i$$

and so if we pick the solution $(\lambda_1, ..., \lambda_{\frac{k}{r}})$ as the coefficients for our linear combination we get the function $\bar{f_i}$ which has the desired property. Note also that $\deg(f_i) \leq (\frac{k}{r} - 1)(r + 1)$ which then implies that $\deg(f_a) \leq k + \frac{k}{r} - 2$ and so because of the inequality established in Theorem 4.0.4 we get optimality.

4.2 Removal of Division Assumption

So far we have assumed that $r + 1|n$. This assumption makes the code construction less flexible so we will modify the construction in order to facilitate arbitrary code length. However, this means that we will no longer obtain optimality in Equation (3.2). We still assume that $r|k$ but this assumption is still nonessential. The code is stated in Construction 4.2.1

Construction 4.2.1
Let $\mathbb{F}$ be a finite field and let $A \subset \mathbb{F}$ such that $|A| = n$ and $n \bmod (r + 1) = s \neq 1$. Now let $w = \lceil \frac{n}{r+1} \rceil$ and let $\mathcal{A} = \{A_1, ..., A_w\}$ be a partition of A such that $|A_i| = r + 1$ for $i = 1, ..., w - 1$ and let $|A_w| = s < r + 1$ be the elements in $A \backslash \cup_{i=1}^{w-1} A_i$. Now let $\Phi_i : \mathbb{F}^{k/r} \to \mathbb{F}_{\mathcal{A}}[x], i = 0, ..., r - 1$ be injective map and assume that Φ_{s-1} is a map whose range is the space of polynomials where $\{f \in \mathbb{F}_{\mathcal{A}}[x] : f(\alpha) = 0 \text{ for } \alpha \in A_m\}$.
We write our message vector in the following way $m = (m_0, ..., m_{r-1})$ where each m_i is a

vector with $\frac{k}{r}$ entries and so $m \in \mathbb{F}^k$. We define the encoding polynomial as:

$$f_m(x) = \sum_{i=0}^{s-1} \Phi_i(m_i, x)x^i + \sum_{i=s}^{r-1} \Phi_i(m_i, x)x^{i-s}h_{A_w}(x).$$

Where h is the annihilator polynomial for A_w defined as $h_{A_w} = \prod_{\alpha \in A_w}(x - \alpha)$. Again the code is defined as $C = \{(f_m(\alpha), \alpha \in A) : m \in \mathbb{F}^k\}$.

Again we must prove that the code constructed is a (n, k, r) LRC code.

Theorem 4.2.2
Construction 4.2.1 defines a (n, k, r) LRC code.

Proof. If an erasure occurs at $f_m(\alpha), \alpha \in \{A_1, ..., A_{w-1}\}$ then it can be recovered by the same recovery procedure described in the proof of Theorem 4.0.4 due to the fact that $f_m(x) \in \mathcal{F}^r_{\mathcal{A}}$. But we must consider the case where $\alpha \in A_w$ In this case we define the polynomial $\delta(x) = \sum_{i=0}^{s-2} \Phi(m, \alpha)x^i$ and note here that $\delta(\alpha) = \Phi(m, \alpha)$ due to $h_{A_m}(x)$ being an annihilator polynomial. Furthermore since $\Phi(m, x) \in \mathbb{F}_{\mathcal{A}}[x]$ we have that $\delta(\beta) = \Phi(m, \beta)$ and from these values $s - 1$ the polynomial $\delta(x)$ can be interpolated. So $\Phi(m, \alpha)$ can be recovered from $s - 1 < r$ symbols. □

Before we prove the bound on the code some modifications and assumptions are necessary they are listed below:

- We assume that $|A| = n$ such that $n \bmod (r + 1) \neq 0, 1$.
- We also assume that $r|k + 1$ however this assumption is simply for ease of notation.
- Let $\mathcal{A} = \{A_1, ..., A_w\}$ be given as in Construction 4.2.1. Let $g(x)$ be a polynomial where $\deg(g) = r + 1$ whose powers $1, g, ..., g^{w-1}$ span $\mathbb{F}_{\mathcal{A}}[x]$. We can assume W.l.o.g that $g(\alpha) = 0$ for $\alpha \in A_m$ if this is not true we can pick the powers of the polynomial $g(x) - g(A_m)$ as a basis for the algebra and these polynomials will satisfy the same condition.
- We split our message $m \in \mathbb{F}^k$ in the following way. We write $m = (m_0, ..., m_{r-1})$ where each m_i is a vector of length $\frac{k+1}{r}$ for $i \neq s - 1$ and m_{s-1} is a vector of length $\frac{k+1}{r} - 1$.

We are now ready to state the modified encoding polynomial.

$$f_m(x) = \sum_{i=0}^{s-2}\left(\sum_{j=0}^{\frac{k+1}{r}-1} m_{i,j}g(x)^j x^i + \sum_{j=1}^{\frac{k+1}{r}-1} m_{s-1,j}g(x)^j x^{s-1} \right) + \sum_{i=s}^{r-1} \sum_{j=0}^{\frac{k+1}{r}-1} m_{i,j}g(x)^j x^{i-s}h_{A_m}(x). \tag{4.2}$$

Once again the codeword for a message m is defined as $(f_m(\alpha), \alpha \in A)$. We are now ready to state Theorem 4.2.3 concerning the minimum distance of the constructed code.

Theorem 4.2.3
The code defined by the encoding function in Equation (4.2) satisfies the following inequality.

$$d \leq n - k - \lceil \frac{k}{r} \rceil + 1 \tag{4.3}$$

Proof. We initially note that the encoding is linear. This implies that the inequality given in Equation (3.6) still holds. So for any polynomial we $f_m(x)$ we have that:

$$\begin{aligned} d_m &\geq n - \max_{f_m(\alpha), m \in F^k} \deg(f_m) = \\ &= n - ((\frac{k+1}{r} - 1)(r+1) + (r-1)) = n - k - \lceil \frac{k}{r} \rceil + 1. \end{aligned}$$

The recovery process is exactly the same as in the proof of Theorem 4.2.2. □

4.3 Redundant Residue Codes

In this section, we will construct an LRC code that can be partitioned into several MDS codes. To do this we first define the following integers:

$$k \leq \sum_i k_i \text{ and } n = \sum_i n_i \text{ where } k_i \leq n_i \text{ for all } i.$$

From this and the Chinese remainder theorem stated in the appendix, we state Construction 4.3.1

Construction 4.3.1
Let A be a subset of $\mathbb{F}$ where $|A| = n$ and consider the following partition $\mathcal{A} = \{A_1, ..., A_t\}$ where $|A_i| = n_i$ for $i = 1, ..., t$. Define the injective map

$$\Phi : \mathbb{F}^k \to \mathcal{F}_{k_1}[x] \times ... \times \mathcal{F}_{k_t}[x].$$

This function takes our message $m \in \mathbb{F}^k$ and maps to the set of encoding functions $(M_1, ..., M_t)$ where M_i is defined from the Chinese remainder theorem. Note also that $\mathcal{F}_{k_i}[x]$ is the space of polynomials where the degree is less than k_i. We define the following annihilator polynomial for $i = 1, ..., t$

$$G_i(x) = \prod_{\alpha \in A_i} (x - \alpha).$$

All these polynomials are pairwise coprime this is because they all have distinct roots. The encoding polynomial is then defined as the unique solution to the equations

$$f_m(x) \equiv M_i(x) \bmod G_i(x).$$

Note here that the $M_i(x)$ are uniquely determined from the injective map Φ Again the code is then defined as the following set of vectors $C = \{(f_m(\alpha), \alpha \in A) : m \in \mathbb{F}^k\}$.

Once again we prove that Construction 4.3.1 yields an LRC code.

Theorem 4.3.2
Construction 4.3.1 yields a (n, k) LRC code with t disjoint local codes C_i where each C_i is an (n_i, k_i) MDS code.

Proof. The encoding from $\mathbb{F}^k$ to $\mathbb{F}^n$ is unique because if it is not then for two messages the polynomial $f_m - f_{m'}$ would have n roots while the degree is less than n. This implies that the mapping from $\mathbb{F}^k$ to $\mathbb{F}^n$ is injective and by Construction 4.0.3 we have that this defines an LRC code because it is defined over the same algebra. For the second part concerning the disjoint local codes consider the set $A_i, i = 1, ..., t$. We note that

$$f_m(x) = h(x)G_i(x) + M_i(x)$$

and since G_i is an annihilator polynomial we get that $f_m(\alpha) = M_i(\alpha)$ for $\alpha \in A_i$. So if we restrict our code to the subset of locations that correspond to A_i we can view the new code as the evaluation of a polynomial with degree less than k_i at n_i points. This implies that the vectors $\{(f_m(\alpha), \alpha \in A_i)|m \in \mathbb{F}^k\}$ form an (n_i, k_i) MDS code. □

Since the code can be viewed as several local MDS codes we have that the $d_m \geq \min_{1 \leq i \leq t}(n_i - k_i + 1)$. This also means that we can construct an LRC code by combining several MDS codes this is shown in Example 4.3.3.

Example 4.3.3.
Let $\mathbb{F}_{17}$ be our base field and consider the evaluation vector for two RS-codes $(3, 4, 7, 2)$ and $(5, 6, 9, 10)$. Also let $k_1, k_2 = 3$ the RS encoding polynomial is thus:

$$\phi_m(x) = \sum_{i=0}^{2} m_i x^i$$

Evaluated at $(3, 4, 7, 2)$ for C_1 and $(5, 6, 9, 10)$ for C_2. By the Chinese Remainder theorem and Construction 4.3.1 we know that the encoding function for the LRC code can be found as the solution to the set of equations:

$$f_m(x) \equiv \phi_m(x) \bmod G_i(x).$$

and thus the code is the set

$$\{(f_m(\alpha), \alpha \in \{3, 4, 7, 2, 5, 6, 9, 10\}), m_1, m_2 \in \mathbb{F}_{17}^3\}$$

We note that the code has locality 3 because each symbol can be recovered from three other symbols in the sets $(3, 4, 7, 2)$ and $(5, 6, 9, 10)$.

Combining codes can be a powerful tool as is shown in the next section where we can construct LRC codes with multiple recovering sets from several LRC codes.

5 | Multiple Recovering Sets

We now extend the definition of locally recoverable codes to include multiple recovering sets. We state the definition in Definition 5.0.1. Note here that we once again assume that $r+1|n$.

Definition 5.0.1
A code $C \subset \mathbb{F}^k$ is locally recoverable with t recovering sets if for every $i \in \{1,...,n\}$ there exists disjoint subsets $A_{i,j} \subset [n] \backslash i, j = 1,...,t$ with size $|A_{i,1}| = r_1, ..., |A_{i,t}| = r_t$ such that for any codeword $c \in C$ the value of c_i is a function of every subset of symbols $\{c_l, l \in A_{i,j}\}, j = 1,...,t$.

We refer to such codes as $LRC(n,k,\{r_1,...,r_t\})$ codes. In order to construct a code with multiple recovering sets we require two partitions of A. So let $\mathcal{A}, \mathcal{A}'$ be two partitions where the size of each set in $\mathcal{A}$ and $\mathcal{A}'$ is $r+1$ and $s+1$ respectively. We can now define the two subspaces:

$$\mathcal{F}^r_{\mathcal{A}} = \oplus_{i=0}^{r-1} \mathbb{F}_{\mathcal{A}}[x]x^i \text{ and } \mathcal{F}^s_{\mathcal{A}'} = \oplus_{i=0}^{s-1} \mathbb{F}_{\mathcal{A}'}[x]x^i \tag{5.1}$$

We note that the dimension of each subspace is:

$$\dim(\mathcal{F}^r_{\mathcal{A}}) = r\frac{n}{r+1},\ \dim(\mathcal{F}^s_{\mathcal{A}'}) = s\frac{n}{s+1} \tag{5.2}$$

This holds because there are $\frac{n}{s+1}$ and $\frac{n}{r+1}$ sets where our function must be constant by the definition of $\mathbb{F}_{\mathcal{A}}$. Now define the following intersection:

$$V_w = \mathcal{F}^r_{\mathcal{A}} \cap \mathcal{F}^s_{\mathcal{A}'} \cap \mathcal{P}_w$$

Where $\mathcal{P}_w = \{p \in \mathbb{F}[x] | \deg(p) < w\}$. V_w is therefore the space of polynomials with degree less than w which also belongs to $\mathcal{F}^r_{\mathcal{A}}$ and $\mathcal{F}^s_{\mathcal{A}'}$. This leads us to Construction 5.0.2.

Construction 5.0.2
First let $A \subset \mathbb{F}$ where $|A| = n$ and let $\mathcal{A}_1, \mathcal{A}_2$ be partitions of A into sets of size $r+1$ and $s+1$ respectively. Assume also that $\dim(\mathcal{F}^r_{\mathcal{A}} \cap \mathcal{F}^s_{\mathcal{A}}) \geq k$ and let w be the smallest integer such that $\dim(V_w) = k$. Let $\Phi : \mathbb{F}^k \to V_w$ be an injective map. We once again construct the code as the set of vectors:

$$\{(\Phi(m,\alpha), \alpha \in A), m \in \mathbb{F}^k\}.$$

Here $\Phi(m, \alpha)$ denotes the evaluation of the polynomial from V_m at α.

An additional assumption is required before we can state the theorem concerning the locality and recovery procedure. We say that two partitions are orthogonal if

$$|X \cap Y| \leq 1, \text{ For all } X \in \mathcal{A}_1, Y \in \mathcal{A}_2$$

With this definition we are now ready to state Theorem 5.0.3.

Theorem 5.0.3
Assume that $\mathcal{A}$ and $\mathcal{A}'$ from Construction 5.0.2 are orthogonal. Then the construction gives an $(n, k, \{r, s\})$ LRC code with distance $d_H \geq n - w + 1$

Proof. For the claim concerning the distance, we do the following argument.

$$\begin{aligned}
&d_H(\Phi(m, x), \Phi(m', x)) \\
&= \text{wt}(\Phi(m, x) - \Phi(m', x)) \\
&= n - |\{\alpha | \Phi(m, x) - \Phi(m', x) = 0\}| \\
&\geq n - \deg(\Phi(m, x) - \Phi(m', x)) \\
&\geq n - w + 1
\end{aligned}$$

Where the last equality holds because the degree of the polynomial is less than w.
Next, we must prove the locality. First, we note that since $\Phi(m) \in \mathcal{F}_{\mathcal{A}}^r$ we have that there exists r polynomials $f_0, ..., f_{r-1} \in \mathbb{F}_{\mathcal{A}}[x]$ such that:

$$\Phi(m, x) = \sum_{i=0}^{r-1} f_i(x) x^i. \tag{5.3}$$

So the recovery procedure is similar to the one presented in the proof of Theorem 4.0.4. Again we state it here for completion. First, consider the indexing of our partitions

$$\mathcal{A} = \{A_{1,r}, ..., A_{a,r}\}, \mathcal{A}' = \{A_{1,s}, ..., A_{b,s}\}$$

Assume that the symbol which must be recovered corresponds to $\Phi(m, \alpha)$ where $\alpha \in A_{i,r}$. We define the decoding polynomial as

$$\delta(x) = \sum_{i=0}^{r-1} f_i(\alpha) x^i$$

Again we have that $\Phi(m, \alpha) = \delta(\alpha)$ and for any $\beta \in A_{i,r}$ we have $\Phi(m, \beta) = \delta(\beta)$ the degree of δ is at most $r - 1$ so δ can be interpolated from the r values $f_m(\beta) = \delta(\beta)$ for $\beta \in A_{i,r} \backslash \alpha$. the exact same argument can be made for the partition $\mathcal{A}'$. □

We will now illustrate the use of this construction in Example 5.0.4.

Example 5.0.4.
We wish to construct a $(10, 4, \{4, 1\})$ LRC(2) code with distance $d_H \geq 2$ so consider the field $\mathbb{F}_{11}$ and let $A = \mathbb{F}_{11}\backslash\{0\}$ we define a partition by considering the cosets of the multiplicative cyclic groups generated 3 and 10.

$$\mathcal{A} = \{\{3, 9, 5, 4, 1\}, \{6, 7, 10, 8, 2\}\}$$

$$\mathcal{A}' = \{\{10, 1\}, \{9, 2\}, \{8, 3\}, \{7, 4\}, \{6, 5\}\}$$

In order to find nice polynomials for each partition we note that $g(x) = x^2$ is a nice polynomial for the partition $\mathcal{A}'$ therefore by Proposition 4.0.1 part 3 we have that $1, g, g^2, ..., g^{w-1}$ forms a basis and so $\mathbb{F}_{\mathcal{A}'}[x] = \langle 1, x^2, x^4, x^6, x^8\rangle$ and similarly we have that $\mathbb{F}_{\mathcal{A}}[x] = \langle 1, x^5\rangle$ since x^5 is a nice polynomial for $\mathcal{A}$. From Proposition 4.0.1 we also get that $\dim(\mathbb{F}_{\mathcal{A}}[x]) = 2$ and $\dim(\mathbb{F}_{\mathcal{A}'}[x]) = 5$. We must now determine the intersection of the spaces $\mathcal{F}^r_{\mathcal{A}}$ and $\mathcal{F}^s_{\mathcal{A}'}$.

$$\begin{aligned}\mathcal{F}^4_{\mathcal{A}} \cap \mathcal{F}^1_{\mathcal{A}'} &= \langle 1, x, x^2, x^3, x^5, x^6, x^7, x^8\rangle \cap \langle 1, x^2, x^4, x^6, x^8\rangle \\ &= \langle 1, x^2, x^6, x^8\rangle.\end{aligned}$$

If $w = 9$ we get that $V_w = \langle 1, x^2, x^6, x^8\rangle$. If we assume that the encoding function $\Phi(x)$ is linear we get that we can write the encoding for a given message $m \in \mathbb{F}^5_{11}$ as

$$\Phi(m, x) = m_0 + m_1x^2 + m_2x^6 + m_3x^8.$$

To illustrate how both recovering sets can be utilized to recover a lost symbol we must first rewrite our encoding polynomial $\Phi(m, x)$ so that it has the form as in Equation 5.3. So we note that

$$\Phi(m, x) = \sum_{i=0}^{3} f_i(x)x^i \tag{5.4}$$

where $f_0 = m_0 + m_4x^5$, $f_1 = m_1$, $f_2 = m_2$, $f_3 = m_3$ where each $f_i \in \mathbb{F}_{\mathcal{A}}[x]$. Similarly for $g_i \in \mathbb{F}_{\mathcal{A}'}$ we get that

$$\Phi(m, x) = \sum_{i=0}^{0} g_i(x)x^i.$$

Where $g_0 = m_0 + m_1x^2 + m_2x^6 + m_3x^8$. Assume we would like to recover the codeword associated with $\Phi(m, 3)$. This can be done in two ways. Either we can access the symbol in $\{3, 9, 5, 4, 1\} \in \mathcal{A}$ or we can access the symbol in $\{8, 3\} \in \mathcal{A}'$.
If we choose $\mathcal{A}$ we find the polynomial $\delta(x)$ with $\deg(\delta) \leq 3$ such that $\delta(9) = \Phi(m, 9)$, $\delta(5) = \Phi(m, 5)$, $\delta(4) = \Phi(m, 4)$ and $\delta(1) = \Phi(m, 1)$. The lost symbol can then be determined as $\delta(1)$.
If we use the partition $\mathcal{A}'$ we would do the following. Find the polynomial $\delta'(x)$ with $\deg(\delta') = 0$ such that $\delta'(8) = \Phi(m, 8)$. Finally, we note that by Theorem 5.0.3 the bound on the distance holds.

The main problem with this construction and theorem is finding orthogonal partitions. However, this problem is alleviated by Proposition 5.0.5 which states necessary and sufficient conditions.

Proposition 5.0.5
Let H, G be two subgroups of a group then the cosets partitions $\mathcal{H}$ and $\mathcal{G}$ are orthogonal iff

$$H \cap G = 1.$$

When the group is cyclic this is equivalent to stating that $\gcd(|H|, |G|) = 1$.

Proof. Start by assuming $H \cap G = 1$ we will show that this implies the partitions are orthogonal. So take a coset of H, Hp, and a coset of G, Gp', and assume two elements $x, y \in Gp' \cap Hp$. Since x, y are in the same coset we have that $Hx = Hy$ and $Gx = Gy$ which is equivalent to requiring $xy^{-1} \in H$ and $xy^{-1} \in G$. By assumption, we have that $H \cap G = 1$ and so $xy^{-1} = 1$ and $x = y$ which implies that all elements in the intersection of the cosets have at most one element.
Now assume that the group is cyclic and let x, y be distinct elements in the group and note that x, y belong to the same coset iff the element xy^{-1} is a $|H|$-th and $|G|$-th root of unity. This is the cast iff $\text{ord}(xy^{-1})$ divides both $|H|$ and $|G|$ which is equivalent to stating $\text{ord}(xy^{-1})|\gcd(|H|, |G|)$ and since x, y are distinct we have that $\text{ord}(xy^{-1}) > 1$ and so $\gcd(|H|, |G|) \neq 1$. □

5.1 LRC Product Codes

One of the more simple ways to construct LRC codes with multiple recovering sets is to utilize product codes this type of construction is described in this section.

Construction 5.1.1
Consider two LRC codes constructed with Construction 4.0.3 with parameters (n_1, k_1, r_1) and (n_2, k_2, r_2) and assume that the two codes C_1 and C_2 are linear. Since they were both constructed using Construction 4.0.3 we have 2 injective maps Φ_1 and Φ_2 which are both linear. We then define the linear map using the tensor product.

$$\Phi = \Phi_1 \otimes \Phi_2 : \mathbb{F}^{k_1 k_2} \to \oplus_{i=0}^{r_1 - 1} \mathbb{F}_{\mathcal{A}_1}[x] x^i \otimes \oplus_{j=0}^{r_2 - 1} \mathbb{F}_{\mathcal{A}_2}[y] y^j$$

So once again Φ sends messages $m \in \mathbb{F}^{k_1 k_2}$ to the set of encoding polynomials $\oplus_{i=0}^{r_1 - 1} \mathbb{F}_{\mathcal{A}_\infty}[x] x^i \otimes \oplus_{j=0}^{r_2 - 1} \mathbb{F}_{\mathcal{A}_2}[y] y^j$. We denote the evaluation of the corresponding polynomial as $\Phi(m, x, y)$ and set this as our encoding polynomial so our code $C = C_1 \otimes C_2$ is the set

$$C = \{(\Phi(m, x, y), (x, y) \in A_1 \times A_2) : m \in \mathbb{F}^{k_1 k_2}\}$$

We state a proposition related to the distance of these product codes.

Proposition 5.1.2
Given two LRC codes (n_1, k_1, r_1) and (n_2, k_2, r_2) Construction 5.1.1 yields a code C with parameters $(n_1 n_2, k_1 k_2, \{r_1, r_2\})$ and distance $d = d_1 d_2$.

Proof. We denote $\mathcal{A}_1 = \sqcup_{j \geq 1} A_j^{(1)}$ and $\mathcal{A}_2 = \sqcup_{j \geq 1} A_j^{(2)}$ as the partition of the evaluation set A_1 and A_2. Let $m \in \mathbb{F}^k$ be given and let $\Phi(m, x, y)$ be the corresponding encoding polynomial. Given some point $(x_0, y_0) \in A_1 \times A_2$ We need to be able to calculate $\Phi(m, x_0, y_0)$ by using r_1 symbols and r_2 symbols from one of the partitions $\mathcal{A}_1$ or $\mathcal{A}_2$. We observe that the polynomial $\Phi(m, x, y_0) \in \oplus_{i=0}^{r-1} \mathbb{F}_{\mathcal{A}_1} x^i$ and so the symbol can be found from the symbols in the set $\{\Phi(m, \alpha, y_0), \alpha \in A_t^{(1)} \backslash x_0\}$ where A_t is the set which contains x_0. To see this we refer to the proof of Theorem 4.0.4.

This argument can be repeated for the polynomial $\Phi(m, x_0, y) \in \oplus_{j=0}^{r_2 - 1} \mathbb{F}_{\mathcal{A}_2}[y] y^j$ and so it the symbol $\Phi(m, x_0, y_0)$ can be recovered from the set $\{\Phi(m, x_0, \beta) \text{ for } \beta \in A_l^{(2)} \backslash y_0\}$ where $A_l^{(2)}$ is the set which contains y_0. So the symbol can be recovered either from $A_t^{(1)} \backslash x_0$ or $A_l^{(2)} \backslash y_0$ which means that the code is an $(n, k\{r_1, r_2\})$ LRC(2) code. Concerning the distance d consider the following:

$$
\begin{aligned}
d &= d_H \Bigg((\Phi(m, x, y)), (\Phi(m', x, y)) \Bigg) \\
&= d_H \Bigg((\Phi(m, x, y_0), x \in A_1), (\Phi(m', x, y_0), x \in A_1) \Bigg) \\
&\cdot d_H \Bigg((\Phi(m, x_0, y), y \in A_2), (\Phi(m', x_0, y), y \in A_2) \Bigg) \\
&= d_1 d_2.
\end{aligned}
$$

Where the first equality holds because we are utilizing a tensor product. □

Let us do an example to illustrate how one can utilize LRC product codes.

Example 5.1.3.
We take the encoding function in Example 3.0.8 with a small modification instead of the nice polynomial $x^7 - 1$ we simply take x^7 which is also a nice polynomial. We now see that the encoding function can be written in the following way based on the vector $b = (1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12)$

$$
f_m(x) = \sum_{i=0}^{11} m_i x^{b_i}.
$$

Given a vector $m \in \mathbb{F}_{29}^{144}, m = (m_{i,j})$ for $i, j = 0, ..., 11$ we can define the encoding function for the product code $C \otimes C$ as:

$$f_m(x, y) = \sum_{i,j=0}^{11} m_{i,j} x^{b_i} y^{b_j}.$$

The set of codewords is then defined as $\{(f_m(x, y)), (x, y) \in A \times A\}$ for all $m \in \mathbb{F}_{29}^{144}$. We need to show that both sets can be used to recover a lost symbol. So assume that the symbol lost corresponds to $f_m(1, 2)$ is lost. We can recover the symbol in two ways:

- Find the polynomial $\delta_1(x)$ with degree less than or equal to 5 such that:

$$\begin{aligned}
\delta_1(7) &= f_m((7, 2) \\
\delta_1(20) &= f_m(20, 2) \\
\delta_1(24) &= f_m(24, 2) \\
\delta_1(23) &= f_m(23, 2) \\
\delta_1(16) &= f_m(16, 2) \\
\delta_1(25) &= f_m(25, 2).
\end{aligned}$$

  Lastly calculate $\delta_1(1) = f_m(1, 2)$.

- Find the polynomial $\delta_2(x)$ with degree less than or equal to 5 such that:

$$\begin{aligned}
\delta_2(14) &= f_m(1, 14) \\
\delta_2(11) &= f_m(1, 11) \\
\delta_2(19) &= f_m(1, 19) \\
\delta_2(17) &= f_m(1, 17) \\
\delta_2(3) &= f_m(1, 3) \\
\delta_2(21) &= f_m(1, 21).
\end{aligned}$$

  Lastly calculate $\delta_2(2) = f_m(1, 2)$.

6 | Proof of Singleton-like Bound

In this chapter, we will prove Theorem 3.0.2. To do this we will need Theorem 6.0.1

Theorem 6.0.1
Let G be a directed graph on n vertices then there exists an induced directed acyclic subgraph G' whose set of vertices U satisfies:

$$|U| \geq \frac{n}{1 + \frac{1}{n}\sum_i d_i^{out}}$$

Where U is the set of vertices of G' and d_i^{out} is the outgoing degree of vertex i.

Theorem 6.0.1 is stated without proof. We will also need Proposition 6.0.2.

Proposition 6.0.2
Let C be an (n, k, r) LRC code then we can write the minimum distance in the following way:

$$d_m = n - \max_{I \subseteq [n]} \{|I| : |C_I| < q^k\}.$$

Proof. Let G be a generator matrix for the code C and let $c \in C$ be given such that $\mathrm{wt}(c) = d_m$ and assume w.l.o.g that $(c_1, ..., c_{d_m}) \neq 0$ and that $(c_{d_{m+1}}, ..., c_n) = 0$ if not then we can simply switch the columns in the generator matrix. Now let $I \subseteq \{d_m + 1, ..., n\}$ we now have that.

$$d_m = \mathrm{wt}(c) = n - |\{d_m + 1, ..., n\}| = n - \max_I |I|$$

and the result follows. □

We will now prove Theorem 3.0.2.

Proof. We construct a directed graph $G = (V, E)$ where V is the set of coordinates $[n]$ of C and say that there is a directed edge from i to j iff $j \in I_i$. Since our code C has locality r this means that the outgoing degree of each vertex i satisfies the following inequality $d_i^{out} \leq r$. From this inequality and Theorem 6.0.1 we get that there exists a subgraph $G' = (U, E')$ whose set of vertices must satisfy:

$$|U| \geq \frac{n}{1 + \frac{1}{n}\sum_i d_i^{out}} \geq \frac{n}{1 + \frac{1}{n}\sum_{i=1}^{n} r} = \frac{n}{1 + r}.$$

Now let i be a coordinate in U without outgoing edges then i is a function of the coordinates $[n]\backslash U$ this is because the coordinate i has locality r. We can iterate on this argument by considering the induced subgraph $G = (U\backslash i, E'')$ this is also an acyclic directed subgraph. Let $i' \in U\backslash i$ with $d_{i'}^{out} = 0$ again this implies that i' is a function of the coordinates $[n]\backslash U$. Since we can do this argument for all coordinates $i \in U$ this implies that any coordinate $i \in U$ is a function of the coordinates in $[n]\backslash U$. This means that the amount of redundancy $n - k$ is bounded from below by $|U| \geq \frac{n}{r+1}$ from this we get the following inequalities.

$$\begin{aligned} n - k &\geq \frac{n}{r+1} \\ 1 - \frac{k}{n} &\geq \frac{1}{r+1} \\ -1 + \frac{k}{n} &\leq -\frac{1}{r+1} \\ \frac{k}{n} &\leq \frac{r+1}{r+1} - \frac{1}{r+1} \\ \frac{k}{n} &\leq \frac{r}{r+1}. \end{aligned}$$

This proves Equation (3.1). We now prove the singleton-like bound given in Equation (3.2). First, note that the minimum distance for LRC codes can, by Proposition 6.0.2, be written in the following way.

$$d_m = n - \max_{I \subseteq [n]} \{|I| : |C_I| < q^k\}. \tag{6.1}$$

Next, we consider a subset of our vertex set $U' \subseteq U$ with size $|U'| = \lfloor \frac{k-1}{r} \rfloor$ and we note that such a subset exists because of the inequality in Equation (3.1) this is shown below.

$$|U| \leq \frac{n}{r+1} \leq \frac{k}{r} \leq \lfloor \frac{k-1}{r} \rfloor.$$

We now define the set N to be the set of coordinates that have at least one incoming edge from a coordinate in U' and we see that $|N| \leq r|U'| = r\lfloor \frac{k-1}{r} \rfloor \leq k - 1$ where the first inequality holds because each coordinate from U to N can be at most connected to r other coordinates in N. Now define N' to be a set with $k-1$ elements defined as the union of N with $k - 1 - |N|$ elements from the set $[n]\backslash(N \cup N')$. We can now establish the following bound on the cardinality of the code restrictions.

$$|C_{N' \cup U'}| = |C_{N'}| \leq q^{k-1}.$$

We can also see that $|N' \cup U'| = k - 1 + \lfloor \frac{k-1}{r} \rfloor$. We can now determine the following lower bound on the restriction of the code:

$$\max_{I \subseteq [n]} \{|I| : |C_I| < q^k\} \geq k - 1 + \lfloor \frac{k-1}{r} \rfloor.$$

This implies from Equation (6.1) that

$$d \leq n - k - \lceil \frac{k}{r} \rceil + 2.$$

□

7 | Cyclic LRC Codes

7.1 Intro to Cyclic Codes

The following section is based on (Calderbank 2008) and (MacWilliams 1983) In this chapter, we will construct cyclic codes with the locality property. Before we state the construction let us define cyclic codes.

Definition 7.1.1
A code C is called cyclic if the shifted codeword

$$S^i(c) = (c_{n-i}, c_{n-i+1}, ..., c_{n-1}, c_0, ..., c_{n-1-1})$$

is still a codeword so $S^i(c) \in C$ for all $i \in \{0, ..., n\}$.

Usually, cyclic codes are defined by ideals of the ring $R_n = \mathbb{F}_q[x]\backslash(x^n - 1)$, which are generated by the polynomials $g(x)$ where $g(x)|x^n - 1$. We note that a codeword in a cyclic code can be described by the following polynomial $c(x) = f(x)g(x)$ where $\deg(f) < k$ and $\deg(g) = t = n - k$ where we associate each coefficient of the polynomial $c(x)$ with a symbol of the codeword c_i.
In order to understand the construction of cyclic LRC codes and the proof of optimality we need Definition 7.1.2 concerning cyclotomic cosets.

Definition 7.1.2
Let n be relatively prime to the field size q the cyclotomic coset of q mod n containing i is defined as:

$$\mathcal{C}_i = \{(i \cdot q^j \bmod n) | j = 0, 1...\}.$$

Furthermore, a subset $\{i_1, ..., i_u\}$ of $\mathbb{Z}_n$ is called a complete set of representatives of cyclotomic cosets if $C_{i_1}, ..., C_{i_u}$ are all distinct and

$$\bigcup_{j=1}^{u} C_{i_j} = \mathbb{Z}_n.$$

There is an important theorem that relates to cyclic codes known as the BCH bound which is a lower bound on the distance. It is stated here without proof.

Theorem 7.1.3
Let C be a cyclic code with generator polynomial $g(x)$ such that for some integers $b \geq 0, \delta \geq 1$ we have

$$g(\alpha^b) = g(\alpha^{b+1}) = ... = g(\alpha^{b+\delta-2}) = 0.$$

Then $d_m \geq \delta$. I.e the minimum distance is bounded from below by δ.

7.2 Cyclic LRC Codes

The following section is based on Tamo et al. (2015). We modify Construction 3.0.3 in the following way: We let the polynomial $g(x)$ be the annihilator of a subgroup of $\mathbb{F}_q^{\cdot}$ and let $n|q-1$. We note that the encoding polynomial given in Construction 3.0.3 can be written in the following way:

$$f_m(x) = \sum_{\substack{i=0, \\ i \neq r \mod (r+1)}}^{\frac{k}{r}(r+1)-2} m_i x^i, \tag{7.1}$$

provided that our message m is indexed as $m = (m_0, ..., m_{k-1})$. We evaluate our function on the set $A = \{1, \alpha, ..., \alpha^{n-1}\}$ where α is a primitive n-th root of unity. We now state Proposition 7.2.1 regarding the cyclic nature of the code.

Proposition 7.2.1
The code defined as $(f_m(1), f_m(\alpha), ..., f_m(\alpha^{n-1}))$ for $m \in \mathbb{F}_q^k$ is cyclic.

Proof. Let $m \in \mathbb{F}_q^k$ and let c be the associated codeword and consider the following cyclic shift of our code word c:

$$S(c_m) = (f_m(\alpha^{n-1}), f_m(1), ..., f_m(\alpha^{n-2})).$$

We wish to find a message $m' \in \mathbb{F}_q^k$ such that the corresponding codeword c' satisfies $c' = S(c)$ for this we consider that the following equations must hold for m'.

$$\begin{aligned} f_{m'}(1) &= f_m(\alpha^{n-1}) \\ f_{m'}(\alpha) &= f_m(\alpha^{n-1}) \\ &\vdots \\ f_{m'}(\alpha^{n-1}) &= f_m(\alpha^{n-2}) \end{aligned}$$

From this set of equations, we get that

$$f_{m'}(1) = f_m(\alpha^{n-1}) \implies \sum m'_i = \sum m_i (\alpha^{n-1})^i$$

therefore set $m'_i = m_i \cdot (\alpha^{n-1})^i$ and note that this is unique because the roots of unity are unique. With our m' defined, we see that

$$f_{m'}(\alpha^j) = f_m(\alpha^{j-1})$$

for a arbitrary $1 \leq j \leq n$. Which is what we needed to show. □

Now that we have proven that the code constructed using the primitive n-th roots of unity as evaluation points are cyclic let us define the set of roots of the polynomial g used to define our code. We let

$$Z = \{\alpha^{i_j}, j = 1, ..., t\}$$

be the roots of the polynomial g. Now the set of unique representatives of cyclotomic cosets in Z with respect to the field $\mathbb{F}_q$ is defined as the *defining set* of zeroes. We now establish Theorem 7.2.2 that defines sets L and D which account for the code's locality and distance.

Theorem 7.2.2
Let α be a primitive n-th root of unity where $n|(q-1)$, let l be an integer where $0 \leq l \leq r$ and let $b \geq 1$ be an integer such that $\gcd(b, n) = 1$ we now define the sets L, D

$$L = \{\alpha^i, i \bmod r + 1 = l\} \text{ and} \tag{7.2}$$

$$D = \{\alpha^{j+sb}, s = 0, ..., n - \frac{k}{r}(r+1)\}. \tag{7.3}$$

Where $\alpha^j \in L$. The cyclic code with the defining set of zeroes $L \cup D$ is an optimal (n, k, r) LRC code over the field $\mathbb{F}_q$.

We need two lemmas and a proposition to prove Theorem 7.2.2. First, we recall the following property of n-th roots of unity.

$$\sum_{i=0}^{n-1} \alpha^i = 0 \text{ if } \alpha \neq 1 \tag{7.4}$$

$$\sum_{i=0}^{n-1} \alpha^i = n \bmod p \text{ if } \alpha = 1.$$

We are now ready to state Lemma 7.2.3.

Lemma 7.2.3
Consider the cyclic code C constructed using the polynomial in 7.1 and the set A, The rows of the generator matrix G are all of the form:

$$(1, \alpha^j, \alpha^{2j}, ..., \alpha^{(n-1)j}). \tag{7.5}$$

For all
$$j \in \{0, 1, ..., \frac{k}{r}(r+1) - 2\} \backslash \{s(r+1) - 1, s = 1, ..., \frac{k}{r} - 1\}.$$
Furthermore, the defining set of zeros of C has the form $R = D \cup \bar{L}$ where:

$$D = \{\alpha^i | i = 1, ..., n - \frac{k}{r}(r+1) + 1\} \tag{7.6}$$

$$\bar{L} = \{\alpha^{n-(\frac{k}{r}-l)(r+1)+1}, l = 1, ..., \frac{k}{r} - 1\}. \tag{7.7}$$

Also, the code C is a cyclic (n, k, r) LRC code with distance $d = n - \frac{k}{r}(r+1) + 2$ with the defining set of zeroes R.

Proof. We initially prove the statement concerning the generator matrix. it is clear that the indices $s(r+1) - 1, s = 1, ..., \frac{k}{r} - 1$ it is clear that $s(r+1) - 1 \bmod r + 1 = r$ so these indices account for the indices that we don't sum over in Equation (7.1) and therefore by removing them we obtain that G is indeed a generator matrix. Next, we prove the statement concerning the zeroes note here that it is sufficient to show that

$$(1, \alpha^j, \alpha^{2j}, ..., \alpha^{(n-1)j}) \cdot (1, \alpha^t, \alpha^{2t}, ..., \alpha^{(n-1)t}) = 0$$

for $t \in R$. Note: When we say $t \in R$ we mean that we take the elements in the representatives which is a set of integers so taking $t \in D$ means taking $t \in \{1, ..., n - k - \frac{k}{r} + 1\}$. Now take the generator element α^j which generates a row of G. We must show that $\alpha^{j+t} \neq 1$ or equivalently that $j + t$ is not a multiple of n. This holds because if $t \in D$ we have that the maximum value that the indices can attain is $j + t = k - 2 + n - k + 1 = n - 1$ and so we get that $j + t$ is not a multiple of n. Now for $t \in \bar{L}$ we have:

$$j + t = j + 1 + n - (\frac{k}{r} - l)(r+1). \tag{7.8}$$

This is because $j = n - (\frac{k}{r} - l)(r+1) + 1$ is a representative of $\bar{L}$. The last two terms on the RHS of Equation (7.8) are both multiples of $r + 1$ so the entire RHS is a multiple of $r + 1$ iff $j + 1$ is a multiple of $r + 1$ however there are no rows in G that would make this possible and so $r + 1 \nmid (j + t)$. Lastly, for the claim concerning the distance we refer to the BCH bound stated in Theorem 7.1.3 when this is applied on the set of zeroes D we get the desired bound. □

Lemma 7.2.4
Let $0 \leq l \leq r$ and define the $\frac{n}{r+1} \times n$ matrix H whose rows are

$$h_m = (1, \alpha^{m(r+1)+l}, \alpha^{2(m(r+1)+l)}, ..., \alpha^{(n-1)(m(r+1)+l)})$$

where $m = 0, ..., \frac{n}{r+1}$. Also, Consider the vector

$$v = (1, 0, ..., 0, \alpha^{l\frac{n}{r+1}}, 0, ..., 0, \alpha^{2l\frac{n}{r+1}}, 0, ..., 0, ..., \alpha^{rl\frac{n}{r+1}}, 0, ..., 0).$$

Where v is the vector where $\mathrm{wt}(v) = r+1$ and where there are $\frac{n}{r+1} - 1$ zeros interspersed between each non-zero entry. All cyclic shifts of the vector v are contained in the row space of H.

Proof. First we show that $av = \sum_{m=0}^{\frac{n}{r+1}} h_m$ where $a = \frac{n}{r+1}$. To see this consider the following for a given coordinate j we have that:

$$\sum_{m=0}^{\frac{n}{r+1}} \alpha^{j(m(r+1)+l)} = \alpha^{lj} \sum_{m=0}^{\frac{n}{r+1}-1} (\alpha^{j(r+1)})^m.$$

From this, we get that we can establish the following based on the values that j takes.

$$j = 0, \;\; \alpha^{lj} \sum_{m=0}^{\frac{n}{r+1}-1} (\alpha^{j(r+1)})^m = n \bmod p$$

$$j = 1, \;\; \alpha^{lj} \sum_{m=0}^{\frac{n}{r+1}-1} (\alpha^{j(r+1)})^m = \alpha^{l} \sum_{m=0}^{\frac{n}{r+1}-1} (\alpha^{(r+1)})^m = 0$$

This follows from the fact that α^{r+1} is a $\frac{n}{r+1}$ primitive root of unity and so by using Equation 7.4 we get the desired result. Expanding this argument for j we get that if j is a multiple of $\frac{n}{r+1}$ we have that the sum $\sum_{m=0}^{\frac{n}{r+1}-1} (\alpha^{j(r+1)})^m = n \mod p$ and therefore the result $av = \sum_{m=0}^{\frac{n}{r+1}} h_m$ follows. This means that the vector av is contained in the row space of H and since the row space H is closed under cyclic shifts then so is the vector av. □

Now the cyclic shifts of the vector v partitions the support of the code into disjoint subsets of size $r+1$ which then become the recovering sets of the local recovering sets of the code. To illustrate this consider the support of codeword v

$$\mathrm{supp}(v) = \{0, \frac{n}{r+1} + 1, 2\frac{n}{r+1} + 1, ..., r \cdot \frac{n}{r+1} + 1\}$$

This set becomes a recovering set for our code. we continue this for all the cyclic shifts of the vector v. We now state the following proposition which describes the locality of the code.

Proposition 7.2.5
Let C be a cyclic code with defining set of zeroes Z and let r be given such that $r+1|n$ if Z contains a coset of the $\frac{n}{r+1}$ roots of unity then C has locality at most r.

Proof. This follows from the fact that v partitions the support of the code and Lemma 7.2.3 and 7.2.4. □

We can also construct cyclic LRC codes with multiple recovering sets. To do this we need a defining set of zeros that contains cosets of subgroups $\frac{k_1}{r_1}, \frac{k_2}{r_2}, ...$, where all the $\frac{k_i}{r_i}$ are pairwise coprime. For example, if $n = 35$ we choose a complete defining set Z which contains the 5-th and 7-th roots of unity the code will have two disjoint recovering sets of size 4 and 6. We are now finally ready to prove Theorem 7.2.2

Proof. The minimum distance for the code C is bounded from below by the BCH bound stated in Theorem 7.1.3 applied on the set of zeroes D. We also know by Proposition 7.2.5. Lastly, the dimension of the code equals $n - |D \cup L| = k$. □

Let's do an example

Example 7.2.6.
First, we define the following integers $n = 21, k = 12, r = 2, q = 64, b = 11$ and note that

$$\frac{n}{r+1} = 7 \text{ and } \frac{k}{r} = 12.$$

We can now define the sets L and D from Theorem 7.2.2.

$$L = \{1, \alpha^3, \alpha^6, \alpha^9, \alpha^{12}, \alpha^{15}, \alpha^{18}\}$$
$$D = \{1, \alpha^1, \alpha^2, \alpha^3\}.$$

If we wish to construct an LRC code from the set of zeroes $R = D \cup L$ we simply find a polynomial $g(x)$ which has R as its set of roots and choose this as our generator polynomial. We then encode a message $m(x)$ by calculating $c(x) = m(x) \cdot g(x)$.

7.3 Subfield Subcodes

In this section, we will describe subfield subcodes of cyclic LRC codes. We first define subfield subcodes.

Definition 7.3.1
Given a code C over a field extension $\mathbb{F}_{q^m}$ the subfield subcode is defined as $C' := C_{\mathbb{F}_q}$ which consists of all the codewords $c \in C$ whose coordinates are in $\mathbb{F}_q$.

Usually, we look at subfield subcodes through the dual of the code using the trace map which we also define.

Definition 7.3.2
Consider the polynomial from T_m from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$ given as:

$$T_m(x) = x + x^q + ... + x^{q^m-1},\ x \in \mathbb{F}_{q^m}.$$

Given a vector $v = (v_1, ..., v_n) \in \mathbb{F}_{q^m}^n$ we use the notation $\mathbf{T}_m(v) = (T_m(v_1), ..., T_m(v_n))$. The trace of a code is obtained as the set

$$T_m(C) = \{\mathbf{T}_m(c), c \in C\}.$$

We state and prove a central theorem concerning the trace map called Delsarte's theorem. It is stated in Theorem 7.3.3 and the proof is adapted from (DELSARTE 1975).

Theorem 7.3.3
For a subfield subcode $C_{|\mathbb{F}_q}$ we have that:

$$(C_{|\mathbb{F}_q})^\perp = T_m(C^\perp).$$

Proof. First, we let $u \in C^\perp$ and $v \in C_{|\mathbb{F}_q}$. We can then do the following:

$$\mathbf{T}_m(u) \cdot v = T(u \cdot v) = T(0) = 0$$

From this we get that $\mathbf{T}_m(u) \in C_{|\mathbb{F}_q}^\perp$ for $u \in C^\perp$ and therefore $T_m(C^\perp) \subseteq C_{|\mathbb{F}_q}^\perp$. Next, take $u \in (T(C^\perp))^\perp$ and $v \in C^\perp$. We know that $C^\perp$ is closed under multiplication by elements in $\mathbb{F}_{q^m}$ so we can do the following

$$T(\lambda(u \cdot v)) = u \cdot \mathbf{T}(\lambda v) = 0$$

From this, we get that $u \in C$ and so the vector u is also in $C_{|\mathbb{F}_q}$. So we get that $C_{\mathbb{F}_q}^\perp \subseteq T(C^\perp)$ which along with $T_m(C^\perp) \subseteq C_{|\mathbb{F}_q}^\perp$ shows that $T_m(C^\perp) = C_{|\mathbb{F}_q}^\perp$. □

Now note that by utilizing the fact the $(C^\perp)^\perp = C$ we can study the code $C_{|\mathbb{F}_q}$ as $(T_m(C^\perp))^\perp$. When studying cyclic LRC codes and their subfield subcodes we can see that $d^\perp(C) = r+1$ this is true because $d^\perp$ is the minimum number of linearly dependent columns in the generator matrix which means that if a symbol is lost we can recover the column and thereby the symbol. We state this fact in the following Proposition 7.3.4.

Proposition 7.3.4

Let C be a cyclic LRC code with locality r then the following holds:

$$d(C^{\perp}) = r + 1.$$

Also, we can see that the subfield subcode in general must have a locality less than r. This is because any coordinate in the dual code is contained in the support of a codevector of weight $\leq r + 1$.
Now let C be a code and $C' = C_{|\mathbb{F}_q}$ as defined above. By Proposition 7.2.5 we have that if Z contains a coset $\{\alpha^i : i \bmod (r+1) = l\}$ of the subgroup generated by α^{r+1} then the code has locality at most r because α^{r+1} is a $\frac{n}{r+1}$ root of unity. Also because of Lemma 7.2.3 we know that the dual code $C^{\perp}$ must contain the vector v which was defined as

$$v = (1, 0, ..., 0, \alpha^{l\frac{n}{r+1}}, 0, ..., 0, \alpha^{2l\frac{n}{r+1}}, 0, ..., 0, ..., \alpha^{rl\frac{n}{r+1}}, 0, ..., 0).$$

We note that $v \in C^{\perp}$ because it is contained in the row space of the parity check matrix. We define the following vector:

$$y := \mathbf{T}_m(\gamma v) \in (C')^{\perp}.$$

Where $\gamma \in \mathbb{F}_{q^m}$ and because the vector v has weight $r + 1$ we get that $\mathrm{wt}(y) \leq r + 1$. For future analysis, we will from now on consider the following subspace of our code $(C')^{\perp}$:

$$V = \langle \mathbf{T}_m(\gamma v), \gamma \in \mathbb{F}_{q^m} \rangle$$

Also from now on we neglect the zeroes of v and view the vector as a vector of length $r+1$. So we write v as the following vector:

$$v = (1, \alpha^{l\frac{n}{r+1}}, \alpha^{2l\frac{n}{r+1}}, ..., \alpha^{rl\frac{n}{r+1}}).$$

There exists a special type of cyclic codes called irreducible cyclic codes they are defined in Definition 7.3.5 and are needed to understand the next theorems.

Definition 7.3.5
Consider the factorization of the polynomial $x^n - 1 = f_1 f_2 ... f_m$ where each f_i is a monic irreducible factor over $\mathbb{F}_q$ the cyclic code generated by the polynomial $\frac{x^n-1}{f_i}$ is called an irreducible cyclic code.

We now state a theorem related to the concept of irreducible codes and the trace function from (Lint 1998).

Theorem 7.3.6
Let $s > 0$ be a given integer and $m = \mathrm{ord}_s(q)$ be the multiplicative order of q modulo s.

Next, let β be a primitive s-th root of unity in $\mathbb{F}_{q^m}$ the set of vectors:

$$V = \{(T_m(\gamma), T_m(\gamma\beta), ..., T_m(\gamma\beta^{s-1})) : \gamma \in \mathbb{F}_{q^m}\} \tag{7.9}$$

is a (s, m) linear irreducible cyclic code over $\mathbb{F}_q$.

Proof. We know that the code is linear by Freshman's dream. Also the code is cyclic because the vector $(T_m(q), T_m(q\beta), ..., T_m(q\beta^{s-1}))$, where $q = \gamma\beta^{-1}$ corresponds to the cyclic shift of the vector $(T_m(\gamma), T_m(\gamma\beta), ..., T_m(\gamma\beta^{s-1}))$ for a given $\gamma \in \mathbb{F}_{q^m}$.
For the claims concerning the dimension and irreducibility consider the following. Because of the requirement that $\text{ord}_s(q) = m$ we have that β does not belong to any subfield of $\mathbb{F}_q$. We know that β is a zero of an irreducible polynomial $h(x) = h_0 + h_1x + ... + h_mx^m$ and thus we can form a parity check equation utilizing this fact

$$\begin{aligned}
&\sum_{i=0}^{m} c_i h_i \\
&= \sum_{i=0}^{m} T_m(\gamma\beta^i)h_i \\
&= \sum_{i=0}^{m} T_m(\gamma\beta^i h_i) \\
&= T_m(\gamma \sum_{i=0}^{m} \beta^i h_i) = 0
\end{aligned}$$

where the linearity of the trace function is utilized. We now see that the polynomial $x^m h(x^{-1})$ becomes a check polynomial and thereby the code V becomes an irreducible (s, m) cyclic code because h is irreducible. □

We first analyze a special case where in our vector v we set $l = 0$. We then know by Theorem 7.2.2 that the defining set of zeroes Z contains the group $G_{r+1} = \langle\alpha^{r+1}\rangle$. Note also that our vector v becomes the vector consisting of the all one vector $v = (1, 1, 1, ..., 1)$ with $r + 1$ entries this vector will span the subspace V. This vector is also contained in $(C')^\perp$ because the identity element is always contained in a subfield and by Lemma 7.2.4 the vector v is contained in the row space of the parity check matrix. We also see that raising an element $\beta \in G_{r+1}$ to the power of q still results in an element in G_{r+1} this is stated below.

$$\beta \in G_{r+1} \implies \beta^q \in G_{r+1}.$$

This holds for all $\beta \in G_{r+1}$. This means that G_{r+1} is a union of cyclotomic cosets. This implies that a cyclic code where $G_{r+1} \subseteq Z$ has locality property.

Example 7.3.7.
Let C' be a $(n = 45, k = 30, d = 4)$ binary cyclic code with zeroes $\{0, 3, 5, 9\}$ in the field $\mathbb{F}_{2^12}$. So the complete set of zeroes are

$$Z = \{3 \cdot 2^j\} \cup \{5 \cdot 2^j\} \cup \{9 \cdot 2^j\} \cup \{0\}$$

for $j = 0, 1....$ We now see that the set of zeroes contains the subgroup G_9. This means that $d^\perp \leq 9$. This is because we know that the vector v is contained in the dual code. Therefore the locality parameter must also satisfy $r \leq 8$ because of the relation $r+1 = d^\perp$. Now consider $(C')^\perp$ this code will have defining set $\{1, 3, 7, 15\}$ this is because the dual code must be generated by the polynomial $x^k h(x^{-1})$ where $h(x) = \frac{x^n-1}{g(x)}$ where $g(x)$ is the generator polynomial for C'. The polynomial $h(x)$ has the roots that are remaining in the set $\{0, 1...n-1\} \backslash Z$ and the inverse of these elements are the roots of our generator polynomial for the dual code $x^k h(x^{-1})$. So we can finally say that the set of roots for $(C')^\perp$ is $\{1, 3, 7, 15\}$.

Now we analyze the case where $l > 0$ and note that we can now split our code into two cases namely where $\gcd(l, r+1) = 1$ and $\gcd(l, r+1) > 1$. However, the case where $\gcd(l, r+1)$ generates a degenerate code where each symbol is repeated. This is because $\alpha^{\frac{n}{r+1}}$ is a $r+1$ root of unity. In further analysis, we, therefore, assume $l = 1$. In order to prove several results concerning the case where $l = 1$ we need Theorem 7.3.8.

Theorem 7.3.8
Let V be a cyclic irreducible code over $\mathbb{F}_q$ as given in Theorem 7.3.6 with length s. Recall that t is the degree of the generator polynomial and assume that $N = \frac{q^m-1}{t}$ is a whole number and assume $\gcd(\frac{q^m-1}{q-1}, N) = 1$ then V is a constant weight code over $\mathbb{F}_q$ with weight $(q-1)q^{m-1}/N$.

The proof of this theorem is beyond the scope of this project but can be found in Ding & Yang (2011). We use Theorem 7.3.8 to prove a bound on the locality and the number of recovering sets.

Proposition 7.3.9
Let $z \geq 1$ be an integer so $2^z - 1 | n$ and let α be an n-th root of unity. Let C be an (n, k) binary cyclic code if the complete defining set Z contains the coset αG_{2^z-1} where $G_{2^z-1} = \langle \alpha^{2^z-1} \rangle$ then the locality parameter satisfies:

$$r \leq 2^{z-1} - 1. \tag{7.10}$$

Also each symbol has at least 2^{z-1} recovering sets A_i with size $2^{z-1} - 1$.

Proof. First, let $\mathbb{F}_{q^z}$ be a subfield of $\mathbb{F}_{q^m}$ and let $T_{m/z} := T_{\mathbb{F}_{q^m}/\mathbb{F}_{q^z}}$ be the trace map which takes elements in $\mathbb{F}_{q^m}$ to elements in the subfield $\mathbb{F}^{q^z}$ and recall that the map $T_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ is

denoted as T_m. We now define two subspaces based on these two maps.

$$V_z = \{(T_z(\gamma), ..., T_z(\gamma\beta^{s-1})) : \gamma \in \mathbb{F}_{q^z}\}.$$

Here $z = \text{ord}_s(q)$ and β is still a s-th root of unity. We also define:

$$V_m = \{(T_m(\gamma), ..., T_m(\gamma\beta^{s-1})) : \gamma \in \mathbb{F}_{q^m}\}.$$

We will prove that $V_m = V_z$ and show that this implies the bound. First, we show that $V_m \subseteq V_z$. We first note that $T_m = T_z \circ T_{m/z}$ is true because of the transitive property of the trace function. The proof is on page 56-57 in (R. Lidl 1989). Using this we can do the following calculation for a given vector $(T_m(\gamma), ..., T_m(\gamma\beta^{s-1})) \in V_m$:

$$\begin{aligned}
&(T_m(\gamma), ..., T_m(\gamma\beta^{s-1})) \\
&= (T_z(T_{m/z}(\gamma)), ..., T_z(T_{m/z}(\gamma\beta^{s-1}))) \\
&= (T_z(T_{m/z}(\gamma)), ..., T_z(T_{m/z}(\gamma)\beta^{s-1})) \in V_z.
\end{aligned}$$

We also need to prove $V_z \subseteq V_m$. Because the function $T_{m/z}$ is surjective we know that there exists a $\gamma' \in \mathbb{F}_{q^m}$ such that $T_{m/z}(\gamma') = \alpha$ for a given $\alpha \in \mathbb{F}_{q^z}\backslash\{0\}$. Now let $(T_z(\delta), ..., T_q(\delta\beta^{s-1})) \in V_z$ we will show that this vector is also contained in V_m. We first construct the following vector:

$$(T_m(\frac{\gamma'\delta}{\alpha}), ..., T_m(\frac{\gamma'\delta}{\alpha}\beta^{s-1})).$$

Where $\frac{\gamma'\delta}{\alpha} \in \mathbb{F}_{q^m}$. we then do the following calculation:

$$\begin{aligned}
&(T_m(\frac{\gamma'\delta}{\alpha}), ..., T_m(\frac{\gamma'\delta}{\alpha}\beta^{s-1})) \\
&= (T_z(T_{m/z}(\frac{\gamma'\delta}{\alpha})), ..., T_z(T_{m/z}(\frac{\gamma'\delta}{\alpha}\beta^{s-1}))) \\
&= (T_z(\frac{\delta}{\alpha}T_{m/z}(\gamma')), ..., T_z(\frac{\delta\beta^{s-1}}{\alpha}T_{m/z}(\gamma'))) \\
&= (T_z(\frac{\delta}{\alpha}\alpha), ..., T_z(\frac{\delta\beta^{s-1}}{\alpha}\alpha)) \\
&= (T_z(\delta), ..., T_z(\delta\beta^{s-1}))
\end{aligned}$$

and so $V_m = V_z$. Now the result follows from Theorem 7.3.8 where we set $q = 2$ used on the set V_z. and the claim concerning the locality and recovering sets follows from Proposition 7.2.5 because $2^z - 1|n$ and because Z contains the coset αG_{2^z-1}. □

We can prove a better bound on the locality using Proposition 7.3.10.

Proposition 7.3.10
Let V be a q-ary (s, m, d) irreducible cyclic code then the minimum distance must satisfy:

$$d_m \leq s(1 - \frac{q^{m-1} - 1}{q^m - 1}).$$

Proof. We first define the linear mapping $T_{m,\gamma} : \mathbb{F}_q^m \to \mathbb{F}_q$ given as $\alpha \to T_m(\gamma\alpha)$ for an element $\gamma \in \mathbb{F}_q^m$ and $\alpha \in \mathbb{F}_q^m$ where $\mathbb{F}_{q^m}$ is viewed as an m-dimensional vector space over $\mathbb{F}_q$. There are q^m linear mappings and these exhaust the set of linear maps from $\mathbb{F}_q^m$ to $\mathbb{F}_q$. This means that for any given γ we can find a vector v_γ such that:

$$T_{m,\gamma}(\alpha) = v_\gamma \cdot \alpha.$$

Where $\cdot$ is the dot product. We now define the following set of indicator random variables

$$X_i = \mathbb{1}(T_{m,\gamma}(\beta^i) = 0).$$

Where $i = 0, ..., s-1$. So this function indicates 1 if the condition $T_{m,\gamma}(\beta^i) = 0$ is met. Now we see that

$$P(X_i) \geq \frac{q^{m-1} - 1}{q^m - 1}.$$

This holds because there are $q^m - 1$ choices for a non-zero vector v_γ which makes the map $T_{m,\gamma}$ non-zero. Also there are at least $q^{m-1} - 1$ possible choices because if for some $i \in \{0, ..., s-1\}$ we have $v_{\gamma,i}\beta^i \neq 0$ we must compensate somehow in the other entries of v_γ. We now realize that this implies the following bound on the expected value of $i \in \{0, ..., s-1\}$:

$$E|\{i : X_i = 1\}| \geq s\frac{q^{m-1} - 1}{q^m - 1}.$$

From this, we can conclude that there exists a γ such that the following bound holds on the weight of a codeword.

$$\mathrm{wt}(T_m(\gamma), T_m(\gamma\beta), ..., T_m(\gamma \cdot \beta^{s-1})) \leq s(1 - \frac{q^{m-1} - 1}{q^m - 1}).$$

□

We now state the proposition concerning the bound on the locality.

Proposition 7.3.11
Let C be an (n, k) cyclic code over $\mathbb{F}_q$ if the complete defining set contains the coset αG_s where α is a primitive n-th root of unity where $s|n$ then

$$r < s(1 - \frac{q^{m-1} - 1}{q^m - 1}).$$

Here m is the multiplicative order of q mod s.

Proof. This follows from the fact that $r(C) < d_m(V)$, which is true because $r(C) = d_m(C^\perp)$ and $d_m(C_{|\mathbb{F}_q}) \geq d_m(C)$. □

So far we have analyzed binary codes we now analyze cyclic ternary codes that are locally recoverable. To begin we state Theorem 7.3.12.

Theorem 7.3.12
Let $N = \frac{q^m-1}{t}$ and assume $\gcd(\frac{q^m-1}{q-1}, N) = 2$ then V is a code of block length t and dimension m and each non-zero codeword has weight:

$$\frac{(q-1)(q^m \pm q^{m/2})}{Nq}$$

and there are $(q^m - 1)/2$ codewords of each weight.

Again this is stated without proof for proof consider Ding & Yang (2011). We now state Proposition 7.3.13 concerning ternary codes.

Proposition 7.3.13
Let C' be an (n, k) ternary cyclic code with defining set Z such that Z contains the coset αG_t for some $t|n$ and where α is n-th degree root of unity. Also let $N = \frac{3^m-1}{t}$ where $m = \text{ord}_3(t)$ and assume that $\gcd(\frac{3^m-1}{2}, N) = 2$ then each symbol of C' has at least $3^{m-1} - 3^{m/2-1}$ recovering sets of size less than $\frac{2(3^m-3^{m/2})}{3N}$.

Proof. We know that the complete defining set contains αG_t and so the $(n = (3^m - 1), k = m)$ irreducible cyclic code V is a shortened code of $(C')^\perp$ because of Theorem 7.3.3 and by Theorem 7.3.12 V contains $\frac{3^m-1}{2}$ codeword with weight $\frac{2(3^m-3^{m/2})}{3N}$. Since the code is cyclic each non-zero coordinate appears as often as non-zero coordinate of the codewords with weight $\frac{2(3^m-3^{m/2})}{3N}$. This is because the cyclic shifts of these codewords are still codewords. This implies that each coordinate is non-zero in $3^{m-1} - 3^{\frac{m}{2}-1}$ codewords with weight $\frac{2(3^m-3^{m/2})}{3N}$. We now note that this must also hold for our vector v and thus we get the result. □

Example 7.3.14.
Let C' be a ternary cyclic code with defining set:

$$Z = \{1, 2, 41\}$$

and length $n = 80$. From the cyclotomic cosets mod 3, we get that the dimension of the code is $k = 80 - 12 = 68$. We now note that the coset αG_{41} is contained in the defining

set of zeros and so we set $t = 40$. Now all the conditions for Proposition 7.3.13 are met so we calculate $m = \text{ord}_3(40) = 4$ and so $N = 2$. So this means that there are $3^{4-1} - 3 = 24$ recovering sets of size less than:

$$\frac{2 \cdot (3^4 - 3^2)}{3 \cdot 2} = 24.$$

For each coordinate.

Appendix 1

In this paper we refer to interpolation one way to do this is using Lagrange interpolation which is defined below. Taken from (Kwak 2011)

Definition .0.15 Lagrange Interpolation
Given a set of nodes $\{\alpha_0, ..., \alpha_k\}$ where $\alpha_i \neq \alpha_j$ for $j \neq i$. We form the Lagrange basis for the polynomial $\{l_0(x), l_1(x), ..., l_k(x)\}$ where each l_j is defined as

$$l_j(x) = \prod_{0 \leq m \leq k, m \neq j} \frac{x - x_m}{x_j - x_m}$$

note here that $l_j(x_j) = 1$ and $l_j(x_i) = 0$ for $i \neq j$. This implies that $l_j(x_i) = \partial_{i,j}$ where $\partial_{i,j}$ is the Kronecker delta function. The Lagrange polynomial for the corresponding evaluation points $\{y_0, ..., y_k\}$ is the unique polynomial

$$L(x) = \sum_{j=0}^{k} y_j l_j(x).$$

This polynomial satisfies $L(x_m) = y_m$.

Furthermore, we define an algebra

Definition .0.16
An algebra over $\mathbb{F}$ is an ordered pair $(A, \star)$ where A is a vector space over $\mathbb{F}$ and $\star$ is a bilinear map

$$\star : A \times A \to A$$

We also construct redundant residue codes. To define redundant residue codes we first note that

$$f_m(x) \bmod x - \alpha^i = f(\alpha^i).$$

Using this fact we can encode our function as

$$M_0, M_1, ..., M_{n-1}$$

where $M_i = f_m(x) \bmod (x - \alpha^i)$ the polynomial $f_m(x)$ can be recovered using the Chinese remainder theorem.

Theorem .0.17 Chinese Remainder Theorem
Let $G_1, ..., G_k \in \mathbb{F}[x]$ be polynomials where $\gcd(G_i, G_j) = 1$ for $i \neq j$ then for any k polynomials $M_1, ..., M_k$ there exists a unique polynomial with $\deg(f) < \sum_{i=1}^{k} \deg(G_i)$ such that

$$f(x) \equiv M_i(x) \bmod G_i(x) \text{ for all } i = 1, ..., t.$$

Note that only $M_1, ..., M_k$ are required to recover $f_m(x)$ so the remaining symbols in the codeword are redundant residues that are included to protect against errors. Codes that are defined this way are called *redundant residue codes*.

A | Bibliography

Calderbank, M. (2008), 'An introduction to linear and cyclic codes', Website for math university of Chicago.
URL: *http://www.math.uchicago.edu/ may/VIGRE/VIGRE2008/REUPapers/Calderbank* 33

DELSARTE, P. (1975), 'On subfield subcodes of modified reed-solomon codes', *IEEE Transactions on Information Theory* **21**(5), 575–576.
URL: *https://ieeexplore.ieee.org/document/1055435* 39

Ding, C. & Yang, J. (2011), 'Hamming weights in irreducible cyclic codes'. 42, 45

Kwak, D. Y. (2011), 'Lagrange interpolation', Korea Advanced Institute of Science and Technology.
URL: *https://mathsci.kaist.ac.kr/ dykwak/Courses/Num365-11/LeadMain-chap4.pdf* 47

Lauritzen, N. (2003), *Concrete Abstract Algebra, From Numbers to Gröbner Bases*, Cambridge University Press. **ISBN**: 078-0-521-82679-2. 5, 12

Li, J. & Li, B. (2013), 'Erasure coding for cloud storage systems: A survey', *Tsinghua Science and Technology* **18**, 259–272. 2

Lint, J. H. (1998), *Introduction to Coding Theory*, Springer Berlin, Heidelberg. **ISBN**: 978-3-540-64133-9. 40

MacWilliams, J. (1983), *theory of error correcting codes*, North Holland. **ISBN**: 9780444851932. 3, 33

R. Lidl, H. N. (1989), *Finite Fields Encyclopedia of Mathematics and its Applications*, Addison-wesley publishing company. **ISBN**: 978-1604779806. 43

Ramkumar, V., Balaji, S. B., Sasidharan, B., Vajha, M., Krishnan, M. N. & Kumar, P. V. (2022), 'Codes for distributed storage', *Foundations and Trends® in Communications and Information Theory* **19**(4), 547–813.
URL: *http://dx.doi.org/10.1561/0100000115* 2

Tamo, I. (2015), 'Locally recoverable code constructions and some extensions', Uploaded to Youtube.
URL: *https://www.youtube.com/watch?v=B9pm1Mk0g2wt=472sab$_c$hannel = SimonsInstitute* 2, 3

Tamo, I. & Barg, A. (2014), 'A family of optimal locally recoverable codes', *IEEE Transactions on Information Theory* **60**(8), 4661–4676.
URL: *https://arxiv.org/abs/1311.3284* 2, 5

Tamo, I., Barg, A., Goparaju, S. & Calderbank, R. (2015), 'A brief history of the development of error correcting codes', *2015 IEEE International Symposium on Information Theory (ISIT)* **10**(3), 1262–1266.
URL: *https://ieeexplore.ieee.org/document/7282658* 34