University of Trento

Department of Mathematics

Ph.D. in Mathematics
Ciclo XXVI

Algebraic methods for the distance of cyclic codes

Matteo Piva

Supervisor:	Prof. Massimiliano Sala
Head of PhD School:	Prof. Francesco Serra Cassano

April, 2014

University of Trento

Department of Mathematics

Ph.D. in Mathematics
Ciclo XXVI

Algebraic methods for the distance of cyclic codes

Ph.D.Thesis of: Matteo Piva

Supervisors: Prof. Massimiliano Sala

Head of PhD School: Prof. Francesco Serra Cassano

April, 2014

Contents

Abstract

Cyclic codes form an interesting part of error correcting codes. The interest in cyclic codes arise from practical reasons, since they are widely employed in many real-life applications, and from theoretical considerations, since they possess a rich algebraic structure (compared to other linear codes) that eases their investigation. Arguably, the most important parameter for a cyclic code is its minimum Hamming distance. The computation of this parameter appears to be a very difficult problem. However, there are efficient methods that allow to lower bound the distance of cyclic codes, taking advantage of the algebraic structure of cyclic codes. This thesis is devoted to study these methods and their theoretical background. In our investigation we do not deal with the aspects concerning the decoding.

In the scientific literature on the subject we can find two main competing approaches in the determination of bounds for the distance, on which we elaborate below.

The very first example of these bounds is the BCH bound, which was proved in 1960 using an argument based on polynomial manipulations leading to a contradiction. We call this approach the polynomial approach. This polynomial approach continued until 1972, where the Hartmann-Tzeng bound was similarly proved, but it was discontinued in 1979, when Blahut and others started investigating bounds based on properties of the Discrete Fourier Transform of codewords. Notably, the most successful result in this direction was the Schaub's bound proposed in 1988. However, the polynomial approach received new strength from the development of the Gröbner basis theory and it started to be used again (in adapted form) since 1996.

It is also possible to divide all known bound in two types, independently from the argument used to prove them. The first type are bounds based only on the information coming from the defining set of a code (e.g., BCH, Hartmann-Tzeng, Roos, Betti-Sala). The second type are bounds which need also the knowledge of the cyclic subcodes (e.g., Schaub, Van Lint-Wilson).

The thesis is divided in two parts. Part I contains preliminary results, part of which are our original contributions. Part II contains the core of our research and presents our main results.

The main results of this thesis can be summarized as follows:

- in Chapter 3 we formally characterize the first type of bounds (that we call `root bounds`) and we give proofs based on the DFT approach for the known bounds, extending also some of them. We also show that the optimal such bound cannot reach the true code distance.

- in Chapter 4 we formally characterize the second type of bounds (that we call `border bounds`) and we give proofs based on the DFT approach for the known bounds, showing an unexpected strong correlation between the two most famous, that is, the Van Lint-Wilson bound and the Schaub bound. We also show that even the optimal such bound cannot reach the true code distance.

- in Chapter 6 we explicitly propose a new root bound, which can be computed in polynomial time, it is provable better than many know-bounds and in outperforms all known bounds for a wide range of computed codes.

- in Chapter 7 we give proofs based on polynomial approach for some known bounds;

- in Chapter 8 we provide an effective algorithm able to compute the optimal root bound in a finite time.

Introduction

Cyclic codes form an interesting part of error correcting codes. The interest in cyclic codes arise from practical reasons, since they are widely employed in many real-life applications, and from theoretical considerations, since they possess a rich algebraic structure (compared to other linear codes) that eases their investigation. Arguably, the most important parameter for a cyclic code is its minimum Hamming distance. The computation of this parameter appears to be a very difficult problem. However, there are efficient methods that allow to lower bound the distance of cyclic codes, taking advantage of their algebraic structure (see for example [BS06, Bos01, HT72, Lev95, Roo83, SWST96, vLW86, ZWZB12, ZB12]). This thesis is devoted to studying these methods and their theoretical background. In our investigation we do not deal with the aspects concerning the decoding.

In the scientific literature on the subject we can find two main competing approaches in the determination of bounds for the distance, on which we elaborate below. The very first example of these bounds is the BCH bound, which was proved in 1960 using an argument based on polynomial manipulations leading to a contradiction. We call this approach the `polynomial approach`. This approach continued until 1972, when the Hartmann-Tzeng bound was similarly proved, but it was discontinued in 1979, when Blahut and others started investigating bounds based on properties of the Discrete Fourier Transform (DFT) of codewords. Notably, the most successful result in this direction was the Schaub bound proposed in 1988. However, the polynomial approach received new strength from the development of the Gröbner basis theory and it has been employed again (in adapted form) since 1996.

It is also possible to divide all known bounds in two types, independently from the argument used to prove them. This division depends on the information precessed as input. The first type are bounds based only on the information coming from the defining set of a code (e.g., BCH, Hartmann-Tzeng, Roos, Betti-Sala). The second type are bounds which need also the knowledge of the cyclic subcodes (e.g., Schaub, Van Lint-Wilson).

We consider only bounds which can be applied to any specific cyclic code rather than bounds which can be applied only to a restricted sub-family of cyclic codes such as the duals of BCH codes (e.g. [CU57, MK93, MM92]) and the quadratic-residue codes ([CS84]).

This thesis is divided in two parts.

Part I contains our preliminaries which consist in classical known results, unpublished contributions and our original results. In particular:

- Chapter 1 up to Section 1.2 recalls well-known notation and some well-known results on linear and cyclic codes ([HP03, PHB98, MS81, PW72]). In Section 1.3 we present some notation from the unpublished preprint [BS07].

- Chapter 2 describes the main tools we use in our study: the DFT and the set $\mathcal{U}$. For Section 2.1 the references are published papers ([BS06, Sch88, Cha98, MS81]), while the material presented in Section 2.2 is taken from the unpublished papers [BS07, Sch88]. Our contribution in this chapter restricts to Proposition 2.1.7, which is however instrumental in obtaining our results of Chapter 3 and 4.

- Chapter 3 treats formally those bounds which depend only on the length and the defining set of a code. We call these `root bounds`. The chapter has several sections. Section 3.1 and Section 3.2 describe the theoretical background of the root bounds. Sections 3.3-3.4 present a sub-family of root bounds, which we call `strict root bounds`, showing how a large part of known classical bounds belong to this family, in particular we prove that strict root bounds include: the BCH bound, the Hartmann-Tzeng bound (even its more general form due to Roos), the Betti-Sala bound and the Boston bounds. Section 3.5 provides the proof that not all root bounds are strict, presenting explicitly bounds which do not belong to the class of strict root bounds, as for example the Roos bound. Section 3.6-3.7 show counterexamples to bounds claimed in the literature, as well as slight generalizations of known bounds. Large part of the material in this chapter comes from [BS07]. Our main improvements are Theorem 3.2.18 and Theorem 3.5.8, which were open problems of [BS07].

- Chapter 4 treats formally those bounds which, beside the defining set and length, need also the knowledge of the defining sets of the cyclic subcodes. We call these `border bounds`. This chapter contains several sections. Section 4.1 and Section 4.2 describe the theoretical background of the border bounds. Sections 4.3 present a sub-family of border bounds, which we call `strict border bounds`. Section 4.4 proves how the most famous border bounds are actually strict border bounds. These include the Van Lint-Wilson shifting bound and the Shaub bound. We are able to prove that these two bounds actually are closely related. Large part of the material in this chapter comes from [BS07]. Our main improvements are Theorem 4.1.19 and Theorem 4.3.10, which were open problems of [BS07] and show that the problem of computing the distance cannot be solved using the length of the code, its defining set and even the defining set of all its cyclic subcodes.

- Chapter 5 recalls well-known results in bounding the minimum distance of cyclic codes using Gröbner bases ([BPW$^+$10, MO09, Cha98, Sal02, Sal07, Aug96]). A brief overview on Gröbner bases is provided in Section 5.1. Two different methods are then presented. In Section 5.2 a method using power sums is explained, while in Section 5.3 we present a method exploiting the generalized Newton identities.

Part II contains our main results.

- In Chapter 6 we explicitly propose a new root bound. Our new bound can be computed in polynomial time, it is provable better than many known bounds (e.g. the Hartmann-Tzeng bound and the Betti-Sala bound) and it outperforms all known polynomial-time bounds for a wide range of computed codes. The chapter contains a preliminary section where two partial results are proved, and a final section with the proof of the main statement. Our proof is based on DFT approach.This result was partially presented in [PS13] and solves an open problem in [BS07].

- In Chapter 7 we follow a polynomial approach based on the generalized Newton identities to provide alternative proofs for all the strict root bounds presented in Section 3.4. Section 7.1 contains our proofs, which use both the classical generalized Newton identities and a new type of identities, obtained manipulating the plain error locator polynomial. We believe that this approach is promising to obtain a mechanical proof of root bounds as discussed in Section 7.2, where we collect some considerations, suggestions and a conjecture for further research on this topic.

- In Chapter 8 we provide an effective algorithm able to compute the optimal root bound in a finite time. Our proof depends heavily on properties of Gröbner bases computed with a field-independent strategy. This chapter is organized in three sections and the main result is contained in the last one, Section 8.3.

The thesis contains also an Appendix where we collect the programs used to compute bounds and some numerical confirmations of our claims.

Part I

Preliminary results

Coding Theory

Given two integers $n \geq 1$ and $N \geq 0$, we denote by $(N)_n$ the remainder of division of N by n. For an integer $N < 0$ we define $(N)_n = n - 1 - (-N)_n$.

If $n \geq 1$ and $N \geq 1$, we denote by $(n, N) = (N, n)$ their greatest common divisor.

Let $n \geq 1$ be a natural number. We denote by $\mathbb{Z}_n^*$ the set $\{h \in \mathbb{N} \mid 1 \leq h \leq n-1,\ (h, n) = 1\}$.

The symbol $\sqcup$ will denote disjoint union.

The symbol $\mathbb{N}$ is used for the set of natural numbers and $\mathbb{Q}$ is used for the rational numbers.

We denote by $\mathbb{F}_q$ the field of q elements, where q is a power of prime p, and with $(\mathbb{F}_q)^n$ the standard n-dimensional vector space over $\mathbb{F}_q$. From now on, when not differently specified, $\mathbb{K}$ is any field (not necessary finite). We indicate as $\overline{\mathbb{F}_q}$ and $\overline{\mathbb{K}}$ the algebraic closure of $\mathbb{F}_q$ and $\mathbb{K}$, respectively.

All the results in this chapter up to Section 1.2 included, are well-known in literature, we use as references [HP03, PHB98, MS81, PW72]. Section 1.3 contains some notation from the unpublished preprint [BS07].

1.1 Linear codes

1.1.1 Basic definitions

Definition 1.1.1. *Let k, n be two integers such that $n \geq k \geq 1$ and let $\phi : (\mathbb{F}_q)^k \mapsto (\mathbb{F}_q)^n$ be an injective map. We say that $C = \mathrm{Im}(\phi)$ is a $[n, k]$-`block code` (or simply `code`) over $\mathbb{F}_q$. If ϕ is linear, then C is called a `linear block code` (or simply `linear code`) of `length` n and `dimension` k over $\mathbb{F}_q$. An element $c \in C$ is called a `word` of C (or `codeword` if C is clear from the context).*

We do not treat in this thesis the case of non-linear codes, so we only say a `code` for a linear block code. The code containing only the zero vector is called the `zero-code`. A code over $\mathbb{F}_2$ is called a `binary` code. When we do not specify the field, we implicitly mean that the code is defined over $\mathbb{F}_q$. Note that if C is an $[n, k]$ code over $\mathbb{F}_q$, then $|C| = q^k$. We denote by $\mathcal{L}_q$ the class of linear codes over $\mathbb{F}_q$ and by $\mathcal{L}$ the union $\mathcal{L} = \cup_q \mathcal{L}_q$.

As subspace of $(\mathbb{F}_q)^n$, a linear code admits a basis. This leads to the definition of a `generator matrix` of a code.

Definition 1.1.2. *Let C be an $[n,k]$-code over $\mathbb{F}_q$. Any matrix G whose rows form a basis for C as a k-dimensional subspace of $(\mathbb{F}_q)^n$ is called a `generator` matrix.*

If G has the form $G = [I_k \mid A]$, where I_k is the $k \times k$ identity matrix, G is called a generator matrix in `standard form`.

In general, there are many generator matrices for a codes, nevertheless any code has a unique generator matrix in standard form. If G is in standard form then the code is called `systematic`.

Let "$\cdot$" be the usual scalar product in $(\mathbb{F}_q)^n$: given $x = (x_0, \dots, x_{n-1})$ and $y = (y_0, \dots, y_{n-1})$, $x \cdot y = \sum_{i=0}^{n-1} x_i y_i$. The orthogonal of a vector subspace of $(\mathbb{F}_q)^n$ is again a vector subspace, so it defines a code.

Definition 1.1.3. *Let C be an $[n,k]$-code over $\mathbb{F}_q$, its `dual code` $C^\perp$ is the set of all n-vectors which are orthogonal to all words of C :*

$$C^\perp = \{\, c' \mid c' \cdot c = 0, \forall c \in C \,\}.$$

We note that $C^\perp$ is an $[n, n-k]$-code over $\mathbb{F}_q$.

Definition 1.1.4. *A `parity-check matrix` H for an $[n,k]$-code is a generator matrix of $C^\perp$.*

From the previous definitions we have easily that G and H are matrices of size, respectively, $k \times n$ and $(n-k) \times n$. To check if an n-vector x belongs to C it is necessary and sufficient to compute $Hx^T = 0$, in fact it holds:

$$\forall\ x \in (\mathbb{F}_q)^n,\ Hx^T = 0 \iff x \in C. \tag{1.1}$$

Definition 1.1.5. *Let x be any vector in $(\mathbb{F}_q)^n$ and let C be an $[n-k]$ code with parity-check matrix H. The vector $s \in (\mathbb{F}_q)^{n-k}$ such that $s = Hx^T$ is called `the syndrome corresponding to` x. The set $\left\{\, Hx^T \mid x \in (\mathbb{F}_q)^n \,\right\}$ is called `the subspace of syndromes` (or simply the `syndromes`).*

Equation 1.1 states that a vector $x \in (\mathbb{F}_q)^n$ is a word of C if and only if the syndrome corresponding to x is zero.

Given two vectors in $(\mathbb{F}_q)^n$, $x = (x_0, \dots, x_{n-1})$, $y = (y_0, \dots, y_{n-1})$, we define the `(Hamming) distance` between x and y as the number of components for which they differ:

$$\mathrm{d}(x,y) = |\,\{\, 0 \le i \le n-1 \mid x_i \ne y_i \,\}\,|.$$

The `(Hamming) weight` of a vector $x \in (\mathbb{F}_q)^n$ is the number, $\mathrm{w}(x)$, of its non-zero components: $\mathrm{w}(x) = \mathrm{d}(x, 0)$.

Definition 1.1.6. *The* `distance of a code` *C is the smallest distances between distinct codewords:*

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

We only write `distance` and `weight` from now on, since other distances and weights will not be considered. By convention, the distance of the zero-code is ∞. If C is a code of length n, dimension k and distance d, we say that C is an $[n,k,d]$ code. It is clear that any $[n,k,d]$ code is also an $[n,k]$ code and that if C is an $[n,k]$ code, then it is also an $[n,k,\mathrm{d}(C)]$ code. Thanks to linearity, it is possible to define the distance of a linear code in another way, as the following result shows.

Proposition 1.1.7. *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$, then*

$$d = \min\{\, \mathrm{w}(c)) \mid c \in C,\ c \neq 0 \,\}\,.$$

Let C be an $[n,k,d]$ linear code over $\mathbb{F}_q$. If D is a vector subspace of C, then we say that D is a (linear) subcode of C. We have $\mathrm{d}(C) \leq \mathrm{d}(D)$.

Definition 1.1.8. *Let C be an $[n,k,d]$ code, we denote by A_i the number of the codewords of weight i. The set of $\{\, A_i \,\}_{0 \leq i \leq n}$ is called the* `weight distribution` *of C. If $A_i = A_{n-i}$ for $0 \leq i \leq n$, then C has* `symmetric weight distribution`*.*

The linearity of C implies that $A_0 = 1$ and $\mathrm{d}(C) = \min\{\, i \geq 1 \mid A_i \neq 0 \,\}$.

1.1.2 Bounds on distance for linear codes

To estimate the distance for a generic linear code is one of the great challenges in coding theory. We state the decision problem for the minimum distance of a linear code.

`Problem:` MINIMUM DISTANCE ($\mathbb{F}_q$)

`Complexity parameter:` $n \in \mathbb{N}$, $n \geq 1$.

`Instance:` An $m \times n$ matrix H over $\mathbb{F}_q$, $m \leq n$, and an integer $0 < w \leq n$.

`Question:` Is there a non-zero vector $x \in (\mathbb{F}_q)^n$ of weight $\leq w$, such that $Hx^T = 0$?

In 1978 Berlekamp, McEliece and van Tilborg [BMvT78] conjectured that the decision problem for the minimum distance of a linear code is NP-complete. The conjecture was solved affirmatively by Vardy in 1997 ([Var97a] and [Var97b]), who also showed that finding the minimum distance is an NP-hard problem. The great interest of mathematicians and coding theory researchers for this question lies in the fact that the distance is a parameter of the performance of a code. We say that a code C has `error correction capability` t if C can correct all errors of weight up to t and there is an error of weight $t+1$ that cannot be corrected by C. Similarly, we say that a code C has `error detection capability` s if C can detect all errors of weight up to s and there is an error of weight $s+1$ that cannot be detected by C. Given a code with distance d, we can center each codeword in a sphere of radius $\left\lfloor \frac{d-1}{2} \right\rfloor$ in such a way that all the spheres are dijoint. Suppose that a codeword is sent. If a vector x is received which is not a codeword, then a naive decoding procedure which we may call `minimum distance` decoding, consists in computing the distance between x and any word of the code. The procedure outputs either the word of the code which is nearest to x, if it exists, or a failure message. If no more than $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors occur, then x is contained in a sphere, and then we can correct it to the right codeword, which is the center of the sphere. An error is detected if and only if the received vector is not a codeword. If more than $d-1$ errors occur, it can happen that a codeword was corrupted in another codeword, thus making the error detection impossible. A large distance for the code implies larger error correction capability and error detection capability, since the spheres are larger. More precisely, the following result holds.

Proposition 1.1.9. *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$, then:*

- *C has detection capability $d-1$*
- *C has correction capability $t = \left\lfloor \frac{d-1}{2} \right\rfloor$*

The following theorem gives an elementary relationship between the weight of a codeword and a parity-check matrix for the code.

Theorem 1.1.10. *Let C be an $[n,k,d]$ code with parity-check matrix H. Let $w \geq 1$. Then for any codeword of weight w there is a linear dependence relation among w columns of H. Conversely, for any linear dependence relation involving w columns of H, there is a non-zero word in C of weight less or equal to w.*

It is possible to extend the previous result to the distance.

Corollary 1.1.11. *A linear code has minimum weight d if and only if its parity-check matrix has a set of d linearly dependent columns and any set of $d-1$ columns is linearly independent.*

The following theorem gives an upper bound for the distance of a code.

Theorem 1.1.12 (Singleton bound)**.** *Let C be an $[n,k,d]$ code. Then*

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

A code which reaches the equality in the Singleton bound is called a `maximum distance separable` code or an `MDS` code.

Another fundamental problem in coding theory is, given n and q^k, to determine a code with maximum d. Alternatively, given n and d, to determine the maximum number $B_q(n,d)$ of codewords in a code over $\mathbb{F}_q$ with length n and minimum distance at least d. We report here some well-known bounds for $B_q(n,d)$. The first result is a consequence of Proposition 1.1.9.

Theorem 1.1.13 (Sphere packing bound)**.**

$$B_q(n,d) \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i} (q-1)^i},$$

where $t = \left\lfloor \frac{(d-1)}{2} \right\rfloor$.

Theorem 1.1.14 (Griesmer bound)**.** *Let* C *be and* $[n,k,d]$ *over* $\mathbb{F}_q$ *with* $k \geq 1$*. Then*

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil .$$

Theorem 1.1.15 (Plotkin bound)**.** *If* $n < \frac{qd}{(q-1)}$, *then*

$$B_q(n,d) \leq \frac{d}{d-(1-\frac{1}{q})n}.$$

Theorem 1.1.16 (Gilbert bound)**.**

$$B_q(n,d) \geq \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i}.$$

Theorem 1.1.17 (Varshamov bound)**.**

$$B_q(n,d) \geq \frac{q^n}{\lceil 1 + \sum_{i=0}^{d-2} \binom{n}{i} (q-1)^i \rceil}$$

1.1.3 Equivalence of linear codes

Despite two codes may be different, they can have many properties in common so that we can consider them as essentially the same code. Suppose for instance to have two codes over $\mathbb{F}_q$, C_1, which is an $[n_1,k_1,d_1]$ code, and C_2, which is an $[n_2,k_2,d_2]$ code, such that $c = (c_1, c_2, \dots, c_{n_1}) \in C_1 \iff \bar{c} = (c_{n_1}, c_1, \dots, c_1) \in C_2$, i.e. C_2 is obtained shifting all words of C_1 to the right. In general, C_1 is different form C_2, but we have $n = n_1 = n_2$, $k = k_1 = k_2$, $d = d_1 = d_2$. Moreover, if $\{ A_i \}_{0 \leq i \leq n}$ is the weight distribution of C_1 and $\{ B_i \}_{0 \leq i \leq n}$ is the weight distribution of C_2, we have $A_i = B_i$ for any $0 \leq i \leq n$. We have just described an example of two `permutation equivalent` codes in preparaion for the formal definition. Let $\mathrm{Sym}(n)$ be the symmetric group on a set of n elements. We extend the action of $\mathrm{Sym}(n)$ to $(\mathbb{F}_q)^n$ as follows: given $x = (x_1, x_2, \dots, x_n) \in (\mathbb{F}_q)^n$, $\sigma \in \mathrm{Sym}(n)$, we define:

$$(x_1, x_2, \dots, x_n)\sigma = (x_{1\sigma^{-1}}, x_{2\sigma^{-1}}, \dots, x_{n\sigma^{-1}}).$$

Definition 1.1.18. *Two linear codes $C_1[n, k_1, d_1]$ and $C_2[n, k_2, d_2]$ are `permutation equivalent` if there is $\sigma \in \mathrm{Sym}(n)$ such that*

$$(x_1, x_2, \dots, x_n) \in C_1 \iff (x_1, x_2, \dots, x_n)\sigma \in C_2.$$

We can express any permutation using a permutation matrix.

Definition 1.1.19. *Given a permutation $\sigma \in \mathrm{Sym}(n)$, its permutation matrix is the $n \times n$ matrix $P_\sigma = (p_{i,j})$ given by*

$$p_{i,j} = \begin{cases} 1 \text{ if } j = i\sigma, \\ 0 \text{ otherwise.} \end{cases}$$

We recall some useful properties of permutation matrices.

Proposition 1.1.20. *Let σ, σ' be two permutation of $\mathrm{Sym}(n)$, $x = (x_1, \dots, x_n)$ a n-tuple of symbols. We have*

i. $P_\sigma P_{\sigma'} = P_{\sigma\sigma'}$

ii. $P_\sigma P_\sigma^T = I_n$, *where I_n is the $n \times n$ identity matrix*

iii. $(x_1, \dots, x_n)\sigma = (x_1, \dots, x_n)P_\sigma$, *where on the right we mean a vector-matrix product.*

Thus, from (iii), if we define $C_1P = \{\, xP \mid x \in C_1 \,\}$, we can say that C_1 and C_2 are permutation equivalent if there is a permutation matrix, P, such that $C_1P = C_2$. More general kinds of equivalence can be considered which preserve the weight of codewords, as we are going to show. We recall that a `monomial matrix` is a square matrix with exactly one non-zero entry in each row and column. A monomial matrix M can be written either in the form DP or the PD, where P is a permutation matrix and D is a diagonal matrix.

Definition 1.1.21. *Let C_1 and C_2 be codes of the same length over $\mathbb{F}_q$, and let G_1 be a generator matrix for C_1. Then C_1 and C_2 are `monomially equivalent` if there is a monomial matrix M so that G_1M is a generator matrix for C_2.*

Two permutation-equivalent codes are also monomially-equivalent codes, but the converse it is not true, except in the binary case, where monomial equivalence and permutation equivalence are precisely the same. We have a more general kind of equivalence when considering also composition with an automorphism of the field $\mathbb{F}_q$. Let γ be an automorphism of $\mathbb{F}_q$, then we can extend the action of γ to $(\mathbb{F}_q)^n$ in the usual way: given $(x_1, \dots, x_n) \in (\mathbb{F}_q)^n$, we write $(x_1, \dots, x_n)\gamma = (x_1\gamma, \dots, x_n\gamma)$. For a code C over $\mathbb{F}_q$, we define $C\gamma = \{ x\gamma \mid x \in C \}$.

Definition 1.1.22. *We say that two codes C_1 and C_2 of the same length over $\mathbb{F}_q$ are* ***equivalent*** *if there is an automorphism γ of $\mathbb{F}_q$ and a monomial matrix M such that $C_2 = C_1 M\gamma$.*

Two monomially-equivalent codes are also equivalent, since it is sufficient to consider as automorphism of $\mathbb{F}_q$ the identity. The converse is true only if $\mathbb{F}_q$ has a prime size. Thus on $\mathbb{F}_2$ all these equivalence are the same. Generally speaking, two equivalent codes has the same weight distribution, but there exist codes with the same weight distribution which are not equivalent. We will see in Subsection 1.2.2 another definition, which is of particular interest for the class of cyclic codes.

1.2 Cyclic codes

In this section we introduce the principal aspects concerning an important subclass of linear codes: cyclic codes. Due to their algebraic structure, many techniques of commutative algebra can be used for the study of these codes, in fact from an algebraic point of view, the investigation of cyclic codes it is equivalent to the investigation of ideals in a suitable principal ideal (commutative) ring. The knowledge of efficient methods for encoding and decoding of cyclic codes boosts their application in real life. However, we do not treat here the vast area of encoding and decoding algorithms, preferring to focus on the problem of bounding distance for cyclic codes.

1.2.1 A first description

Given an n-vector $c = (c_0, \dots, c_{n-1})$ we consider its right shift

$$\mathrm{sh}(c) = (c_{n-1}, c_0, \dots, c_{n-2})$$

which is again an n-vector with the same field of coefficient of c. We adopt the usual notation $\mathrm{sh}^i(c)$ to indicate the i-th right shift of c, i.e.:

$$\mathrm{sh}^i(c) = (c_{n-i}, \dots, c_{n-1}, c_0, \dots, c_{n-i-1}).$$

We clearly have $\mathrm{sh}^0(c) = \mathrm{sh}^n(c) = c$ and $\mathrm{sh}^i(c) = \mathrm{sh}^{(i)_n}(c)$.

Definition 1.2.1. *Let C be an $[n,k,d]$ code such that*

$$\forall c \in C, \mathrm{sh}(c) \in C.$$

Then we call C a `cyclic code`.

Thus cyclic codes are invariant with respect to shifts. To get an algebraic description, we can view a vector $c = (c_0, \dots, c_{n-1}) \in (\mathbb{F}_q)^n$ as a polynomial $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ in $\mathbb{F}_q[x]$ of degree at most $n-1$. For a word $c \in C$ we use interchangeably the vector notation or the polynomial notation $c(x)$. The fact that a cyclic code is invariant under cyclic shifts implies that if $c(x)$ is in the code, then $xc(x)$ is so, if we consider the multiplication modulo $x^n - 1$. Let $R_n = \mathbb{F}_q[x]/\langle x^n - 1\rangle$ be the ring consisting of residue classes of $\mathbb{F}_q[x]$ modulo $x^n - 1$. Each polynomial of degree at most $n-1$ belongs to a different residue class and we take this polynomial as representative. Actually, R_n is an algebra over $\mathbb{F}_q$. The consideration above suggests an obvious isomorphism $(\mathbb{F}_q)^n \mapsto \{f \in \mathbb{F}_q[x] \mid \deg(f) \leq n-1\}$ as vector spaces over $\mathbb{F}_q$, given by $(c_0, \dots, c_{n-1}) \mapsto c_0 + \ldots + c_i x^{i-1} + \ldots + c_{n-1}x^{n-1}$. Thanks to this characterization, we can see linear codes of length n as subsets of R_n, in particular, cyclic codes in $(\mathbb{F}_q)^n$ correspond to ideal in R_n, as the following theorem states.

Theorem 1.2.2. *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$, then C is cyclic if and only if C is an ideal of R_n.*

Since R_n is a principal ideal ring, any ideal, C, is generated by an element $g(x) \in R_n$, $C = \langle g(x)\rangle$. If we require that $g(x)$ is monic and of lowest degree, then it is unique. Such polynomial g is called the `generator polynomial` of C. Note that $g \mid (x^n - 1)$ in $\mathbb{F}_q[x]$. The next theorem summarizes this and other properties of cyclic codes.

Theorem 1.2.3. *Let C be a non-zero ideal in R_n i.e., a cyclic code of length n.*

(a) There is a unique monic polynomial $g(x)$ of minimal degree in C.

(b) $C = \langle g(x)\rangle$, i.e. $g(x)$ is a generator polynomial of C.

(c) $g(x)$ is a factor of $x^n - 1$.

(d) If the dimension of C is k, then $\deg(g) = n - k$.

(e) Any $c(x)$ can be written uniquely as $c(x) = f(x)g(x)$ in $\mathbb{F}_q[x]$, where $f(x) \in \mathbb{F}_q[x]$ has degree less than k.

(f) If $g(x) = g_0 + g_1x + \cdots + g_{n-k}x^{n-k}$*, then a generator matrix for* C *is*

$$\begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{n-k-1} & g_{n-k} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & g_0 & g_1 & g_2 & \dots & g_{n-k} \end{pmatrix}$$

We can then replace our first definition of cyclic code, using the result of Theorem 1.2.3.

Definition 1.2.4. *Let* C *be an* $[n, k, d]$ *linear code in* $(\mathbb{F}_q)^n$*. We say that* C *is a cyclic code if there is a monic polynomial* $g_C \in \mathbb{F}_q[x]$ *s.t.* $g_C|(x^n - 1)$ *and* $C = \{g_C f \mid f \in \mathbb{F}_q[x], \deg(f) \le k - 1\}$*. The polynomial* g_C *is called the* `generator polynomial` *of* C*,* $\deg(g_C) = n - k$*, and we write* $C = \langle g_C \rangle$*.*

To simplify the notation, we usually write g to indicate the generator polynomial instead of g_C, when C is clear. Vice versa, any monic $g \in \mathbb{F}_q[x]$ s.t. $g|(x^n - 1)$ generates a cyclic code of dimension $k = n - \deg(g)$. We denote by $\mathcal{C}_{q,n}$ the class of all cyclic codes of length n over $\mathbb{F}_q$, by $\mathcal{C}_n$ the class of all cyclic codes of length n, by $\mathcal{C}_q$ the union $\cup_{(n,q)=1}\mathcal{C}_{q,n}$ and by $\mathcal{C}$ the whole class $\mathcal{C} = \cup_q \mathcal{C}_q$.

Let C be a cyclic code of length n with generator polynomial g. Since g is a divisor of $x^n - 1$, we can define the `check polynomial` of C as $h(x) \in R_n$ such that $h(x) = (x^n - 1)/g(x)$. Note that $h(x)$ and $g(x)$ are zero divisors in the ring R_n. Using the check polynomial it is easy to decide if $c(x) \in R_n$ belongs to the code C. In fact:

$$c(x) \in C \iff c(x) = f(x)g(x) \iff c(x)h(x) = f(x)g(x)h(x) = 0 \text{ in } R_n.$$

We have:

Proposition 1.2.5. *Let* $h(x)$*,* $g(x)$ *be, respectively, the check polynomial and the generator polynomial of the cyclic code* C*. The dual code* $C^\perp$ *is cyclic with generator polynomial*

$$g^\perp(x) = x^{\deg(h)} h(x^{-1}).$$

A generator matrix for $C^\perp$ *is*

$$H = \begin{pmatrix} 0 & \dots & 0 & h_k & \dots & h_1 & h_0 \\ \dots & 0 & h_k & \dots & h_1 & h_0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_k & \dots & h_1 & h_0 & 0 & \dots & 0 \end{pmatrix}$$

Corollary 1.2.6. *Let $h(x)$ be the check polynomial of the cyclic code C. Then the code $C_h = \langle h \rangle$ and $C^{\perp}$ are permutation equivalent. The equivalence permutation is $i\sigma = n - i + 1$.*

From now on, during the study of cyclic codes we make the basic assumption that $(n, q) = 1$ (the other case is studied in [vL95] and [CMSvS91]). The first reason for this assumption is that in this way the polynomial $x^n - 1 \in \mathbb{F}_q[x]$ has distinct roots in its splitting field. The n-th roots of unity are partitioned in q-cyclotomic cosets modulo n. This leads us to a very useful characterization of a cyclic code from the roots of its generator polynomial. Another reason will be clear in Section 2.1 when we introduce the Discrete Fourier Transform.

Since $\mathbb{F}_q$ is not algebraically closed, it is not guaranteed that the roots of $x^n - 1$ belong to $\mathbb{F}_q$. The smallest field which contains the roots of $x^n - 1$ is called `the splitting field` of $x^n - 1$ (over $\mathbb{F}_q$), which we denote as $\mathbb{F}$. We summarize some results on the splitting field.

Theorem 1.2.7. *Let $\mathbb{F}_q$ be a finite field and $\mathbb{F}$ be the splitting field of $x^n - 1$ over $\mathbb{F}_q$. Let $(n, q) = 1$, then*

- *there are a prime p and a positive integer r such that $q = p^r$;*
- *there exist are positive integers m and M such that $\mathbb{F} = \mathbb{F}_{q^m} = \mathbb{F}_{p^M}$*
- *there is an element $\alpha \in \mathbb{F}$ such that*

$$(x^n - 1) = \prod_{i=0}^{n-1} (x - \alpha^i).$$

Such element is called ***`a primitive n-th root of unity`.***

Let $\mathbb{F}$ be the splitting field of $x^n - 1$ over $\mathbb{F}_q$ and let α be a primitive n-th root of unity in $\mathbb{F}$. If g is the generator polynomial of an $[n, k, d]$ cyclic code, then $g \mid (x^n - 1)$ and its roots are a subset of $\{ \alpha^i \mid 0 \leq i \leq n - 1 \}$, which we can collect in a set.

Definition 1.2.8. *Let g be the generator polynomial of an $[n, k, d]$ code over $\mathbb{F}_q$ and α be a primitive n-th root of unity in $\mathbb{F}$, the splitting field of $x^n - 1$. We denote by $S_{C,\alpha}$ the set*

$$S_{C,\alpha} = \{0 \leq i \leq n - 1 \mid g(\alpha^i) = 0\}.$$

$S_{C,\alpha}$ is called the `complete defining set` of C w.r.t. α.

We define the `cyclotomic coset` mod n over $\mathbb{F}_q$ (or, briefly, the q-`cyclotomic coset`) which contains i as $C_i = \{ i, iq, iq^2, \ldots, iq^{m_i - 1} \}$, where $iq^{m_i} \equiv i \mod n$. We can collect the integers modulo n into disjoint q-cyclotomic classes. Recalling that if $g(\alpha^i) = 0$ then also $g(\alpha^{qi}) = g(\alpha^i)^q = 0$, we obtain that the complete defining set of C is a collection of $q-$cyclotomic cosets, i.e. for some $s \geq 1$:

$$S_{C,\alpha} = \bigsqcup_{j=1}^{s} C_{i_j}, \qquad\qquad C_{i_j} = \left\{ i_j, i_j q, i_j q^2, \ldots, i_j q^{m_{i_j}-1} \right\}.$$

Thanks to Theorem 1.2.3 there is a one-to-one correspondence between non-zero cyclic codes and the divisors of $x^n - 1$ different from $x^n - 1$ itself. Moreover, once α is fixed, we have a one-to-one correspondence between irreducible factors of $x^n - 1$ and q-cyclotomic cosets modulo n. These correspondences lead to the following proposition.

Proposition 1.2.9. *The number of non-zero cyclic codes of R_n is $2^r - 1$, where r is the number of q-cyclotomic cosets modulo n.*

Definition 1.2.10. *Let C be a cyclic code. A linear subcode C' of C that is cyclic will be called a* **`cyclic subcode`***. In this case we will write $C' < C$ if C' is not zero.*

Given a cyclic code C with generator polynomial g, it is not difficult to count the number of its proper cyclic subcodes, using the following result.

Proposition 1.2.11. *Let C_1 and C_2 be cyclic codes over $\mathbb{F}_q$ with generator polynomial $g_1(x)$ and $g_2(x)$, respectively. Then $C_1 < C_2$ if and only if $g_2(x) \mid g_1(x)$.*

Thus, if $x^n - 1$ has r irreducible factors and g has s (obviously $s \leq r$) irreducible factors, then C has exactly

$$\sum_{i=1}^{r-s} \binom{r-s}{i} = 2^{r-s} - 1$$

non-null cyclic subcodes (including C itself).
We have that a cyclic code of length n is defined by its complete defining set. In fact:

$$c \in C \iff c(\alpha^i) = 0 \text{ for any } i \in S_{C,\alpha}$$

Note that this fact it is not true if we drop the assumption $(n, q) = 1$. It follows that if $S_{C,\alpha} = \{ i_1, i_2, \ldots, i_{n-k} \}$ is the complete defining set of a cyclic code C of length n, the matrix

$$H = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \ldots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \ldots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{i_{n-k}} & \alpha^{2i_{n-k}} & \ldots & \alpha^{(n-1)i_{n-k}} \end{pmatrix} \tag{1.2}$$

is a parity-check matrix for C. In fact:

$$Hc^T = \begin{pmatrix} c(\alpha^{i_1}) \\ c(\alpha^{i_2}) \\ \vdots \\ c(\alpha^{i_{n-k}}) \end{pmatrix} = 0 \iff c \in C$$

Remark 1.2.12. We note that the entries of matrix H in (1.2) are in $\mathbb{F}$ rather than $\mathbb{F}_q$. clearly C is the null space of H over $\mathbb{F}_q$. H can also be used as a parity-check matrix for a cyclic code C' over $\mathbb{F}$, with the same defining set. We have $C = C'_{|\mathbb{F}_q} = C' \cap (\mathbb{F}_q)^n$ and we say that C is the `subfield subcode` of C' with respect to $\mathbb{F}_q$. Choosing a basis for $\mathbb{F} = \mathbb{F}_{q^m}$ (see Theorem 1.2.7) as vector space over $\mathbb{F}_q$, we obtain an $m(n-k) \times n$ matrix, $H_{|\mathbb{F}_q}$, with entries in $\mathbb{F}_q$. A parity-check matrix for $C'|_{\mathbb{F}_q}$ can be obtained from $H_{|\mathbb{F}_q}$ by deleting the linear dependent rows.

We conclude this section recalling two remarkable families of cyclic codes: `BCH` codes (see [BRC60]) and `Reed-Solomon` codes.

Definition 1.2.13. *We say that a cyclic code $C \in \mathcal{C}_{q,n}$ is a (narrow-sense) BCH code with designed distance δ, if there is an n-th root of unity α over $\mathbb{F}_q$ s.t. $\{1, 2, \ldots, \delta - 1\} \subset S_{C,\alpha}$ and C is the largest code in $\mathcal{C}_{q,n}$ possessing this property.*

Definition 1.2.14. *Given an integer $m \geq 2$, a prime p and an integer $p^m - 1 \geq \delta \geq 1$, let $n = p^m - 1$ and $q = p^m$. Consider polynomial $g \in \mathbb{F}_q[x]$,*

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

where α is a primitive element of $\mathbb{F}_q$. The `Reed-Solomon` code of designed distance δ over $\mathbb{F}_q$ is the cyclic code generated by g.

The Reed-Solomon codes form a sub-class of the BCH codes. Let S be the complete defining set of a cyclic code. Suppose that q and n are known, let $T \subset S$ be such that any cyclotomic class in S has at least an element in T, then T is usually called a `defining set`, since the knowledge of T provides the knowledge of S. In the following, when we write "defining set", we actually mean the complete defining set, unless specify otherwise.

1.2.2 A second description

Previously, we have seen how we can describe a cyclic code as an ideal of R_n, now we introduce a second description of R_n which is often helpful. Let α be a primitive n-th root of unity in $\mathbb{F}$, then $G^* = \{ 1, \alpha, \ldots \alpha^{n-1} \}$ is a subgroup of $\mathbb{F}^*$, the multiplicative group of $\mathbb{F}$. In particular if the length n is equal to $p^m - 1$ for some prime p and some integer $m \geq 1$, then $G^* = \mathbb{F}^*$ (in this case we say that code is `primitive`).

We have already seen how a vector of $(\mathbb{F}_q)^n$ can be identified with an element of R_n by:

$$(c_0, c_1, \dots, c_{n-1}) \mapsto c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}.$$

Similarly, there is another natural way to represent a vector in $(\mathbb{F}_q)^n$, adopting a group algebra point of view:

$$(c_0, c_1, \dots, c_{n-1}) \mapsto c_0\alpha^0 + c_1\alpha^1 + \cdots + c_{n-1}\alpha^{n-1}.$$

We have an isomorphism between R_n and a group algebra, as we are going to explain. Let us indicate with $\mathbb{F}_qG^*$ the group algebra $\mathbb{F}_q[\{\ G^*, \cdot\ \}]$ of the multiplicative group G^* over $\mathbb{F}_q$, consisting of the formal sums:

$$\sum_{i=0}^{n-1} c_i\alpha^i, \quad c_i \in \mathbb{F}_q.$$

Addition and scalar multiplication are component-wise and multiplication is given by multiplication in G^*:

$$\begin{aligned} \sum_{i=0}^{n-1} x_i\alpha^i + \sum_{i=0}^{n-1} y_i\alpha^i &= \sum_{i=0}^{n-1}(x_i + y_i)\alpha^i, \\ \lambda\sum_{i=0}^{n-1} x_i\alpha^i &= \sum_{i=0}^{n-1}(\lambda x_i)\alpha^i, \quad \lambda \in \mathbb{F}_q \\ \sum_{k=0}^{n-1} x_i\alpha^i \cdot \sum_{j=0}^{n-1} y_i\alpha^i &= \sum_{i=0}^{n-1}\left(\sum_{kj=i \mod n} x_k y_j\right)\alpha^i. \end{aligned}$$

With this assumptions we have that the maps $\psi : R_n \mapsto \mathbb{F}_qG^*$

$$\psi\left(\sum_{i=0}^{n-1} c_i x^i\right) = \sum_{i=0}^{n-1} c_i\alpha^i$$

is an isomorphism between the algebras R_n and $\mathbb{F}_qG^*$. Any ideal in $\mathbb{F}_q * G$ is called a `group algebra code` and it is the image by ψ of an ideal in R_n. Thus any cyclic code corresponds to a group algebra code and vice versa.The shift of a codeword $\sum_{i=0}^{n-1} c_i\alpha^i$ is the codeword $\sum_{i=0}^{n-1} c_i\alpha^{i+1}$.

1.2.3 Naturally equivalent cyclic codes

Given a finite field $\mathbb{F}_q$ and an $n \geq 1$ s.t. $(n,q) = 1$, there can be many primitive n-th roots of unity. The following definition allows us to treat formally the choice of a primitive root.

Definition 1.2.15 ([BS07]). *We denote by $\mathcal{S}$ the subset of $\mathbb{N} \times \mathbb{N}$ s.t.*

$$(q,n) \in \mathcal{S} \qquad \Longleftrightarrow \qquad q = p^m, \, p \text{ is a prime}, \, m \geq 1, \, n \geq 1, \, (n,p) = 1 \, .$$

We denote by $\mathcal{Z}$ the class of all functions

$$\zeta : \mathcal{S} \mapsto \sqcup_{p \text{ prime}} \overline{\mathbb{F}_p}$$

s.t. $\zeta(p^m, n) \in \overline{\mathbb{F}_p}$ is a primitive n-th root of unity over $\mathbb{F}_p$.

The following proposition comes from elementary field theory.

Proposition 1.2.16. *Let α be a primitive n-th root of unity over $\mathbb{F}_p$ and $m \geq 1$. Then α is a primitive n-th root of unity over $\mathbb{F}_{p^m}$.*

Many notions of code equivalence are known in coding theory, and in Section 1.1 we mentioned some of them. Here we are going to describe a special case of equivalence, that we call `natural`. We have seen that, once fixed a finite field $\mathbb{F}_q$, a length n and primitive n-th root of unity α, the complete defining set with respect to α determines uniquely the cyclic code. However, since the n-th roots of unity form a cyclic group, G^*, of order n, many different choices for a primitive root can be done. We recall that, if α is a primitive n-th root of unity, then the set of all primitive roots of unity is $\{\, \alpha^s \mid 0 \leq s \leq n-1, \ (s,n) = 1 \,\}$, with cardinality $\varphi(n)$, where φ denotes the well-known Euler function. Given a cyclic code C, different choices for α give different complete defining sets and thus different codes, which are actually "essentially the same" code.

Definition 1.2.17. *Let $C_1, C_2 \in \mathcal{C}_{q,n}$. We say that C_1 and C_2 are* ***`naturally equivalent`*** *if there are two n-th roots of unity over $\mathbb{F}_q$, α and β, s.t.*

$$S_{C_1,\alpha} = S_{C_2,\beta} \, .$$

We have that two naturally equivalent cyclic codes are also permutation equivalent. Let us consider the group algebra $\mathbb{F}_q G^*$. The permutations $\alpha^i \sigma_s = \alpha^{is}$ with $(s,n) = 1$ forms a group $\mathcal{G}$ of automorphisms of G^*. Thus $\mathcal{G}$ permutes the coordinates of $\mathbb{F}_q G^*$. However note that now we consider $\mathrm{Sym}(n)$ as the permutation group acting on $\{0, \dots, n-1\}$ rather than $\{1, \dots, n\}$.

Example 1.2.18. Let us consider $\mathbb{F}_q = \mathbb{F}_3$ and $n = 11$. We have cyclotomic classes:

$$(0) = \{\, 0 \,\} \qquad (1) = \{\, 1, 3, 4, 5, 9 \,\}\,, \qquad (3) = \{\, 2, 6, 7, 8, 10 \,\}$$

Accordingly to the factorization of $x^{11} - 1$ in $\mathbb{F}_3[x]$:

$$x^{11} - 1 = (x + 2)(x^5 + 2x^3 + x^2 + 2x + 2)(x^5 + x^4 + 2x^3 + x^2 + 2)$$

Let α be a primitive $11-$th root of unity with minimal polynomial $x^5 + 2x^3 + x^2 + 2x + 2$, β be a primitive n-th root of unity such that $\beta = \alpha^2$. Note that the minimal polynomial of β is $x^5 + x^4 + 2x^3 + x^2 + 2$. Consider the cyclic code C_1 with defining set $S_{C_1,\alpha} = \{\, 1, 3, 4, 5, 9 \,\}$ and the cyclic code C_2 with $S_{C_2,\alpha} = \{\, 2, 6, 7, 8, 10 \,\}$. Then C_1 is naturally equivalent to C_2, because $S_{C_1,\alpha} = S_{C_2,\beta}$. The permutation $\sigma_2 \in \mathcal{G}$ defined by $\alpha^i\sigma_2 = \alpha^{2i}$ induces a permutation in $\mathrm{Sym}(n)$, which we still call σ_2, with abuse of notation, acting as $i\sigma_2 = (2i)_n$. As a product of cycles in $\mathrm{Sym}(n)$, we have $\sigma_2 = (0)(1\ 2\ 4\ 8\ 5\ 10\ 9\ 7\ 3\ 6)$. The permutation matrix associated to σ_2 is

$$P_{\sigma_2} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0 \end{pmatrix}$$

So, we have $C_2 = C_1P_{\sigma_2}$ (see Section 1.1.3).

Remark 1.2.19. Two permutation equivalent codes not necessarily are naturally equivalent, but only those of the form $C_1 = C_2\sigma_s$ with $(s, n) = 1$.

A classical result on cyclic codes can be rephrased in our context as follows.

Theorem 1.2.20 ([BS07])**.** *Let C_1 and C_2 be naturally equivalent cyclic codes. Then*

$$\mathrm{d}(C_1) = \mathrm{d}(C_2)\,.$$

Furthermore, let C_1 be in $\mathcal{C}_{q,n}$. Let α and β be primitive n-th roots of unity. Then there is a unique cyclic code C_2 in $\mathcal{C}_{q,n}$ s.t.

$$S_{C_1,\alpha} = S_{C_2,\beta}\,.$$

From the defining set of a code it is immediate to find the defining sets of its naturally equivalent codes, as follows. Let $C \in \mathcal{C}_{q,n}$ and $\alpha = \zeta(q, n)$, for a $\zeta \in \mathcal{Z}$. Let $S_{C,\alpha} = \{i_1, \ldots, i_r\}$. Then for any $l \in \mathbb{Z}_n^*$ we can construct a set $S_l = \{j_1, \ldots, j_r\} \subset \{0, \ldots, n-1\}$, where $j_h = (li_h)_n$ for any h. So we can rephrased another classical result as follows.

Theorem 1.2.21 ([BS07]). *For any code D naturally equivalent to C there is an $l \in \mathbb{Z}_n^*$ s.t. $S_{D,\alpha} = S_l$.*
Conversely, for any $l \in \mathbb{Z}_n^*$ there is a code D naturally equivalent to C s.t. $S_{D,\alpha} = S_l$.

Remark 1.2.22. Observe that defining sets of naturally equivalent codes do not depend on the underlying field, but only on n and the defining set of one code, since using different α gives rise to the same set of defining sets.

1.3 Terminology for general distance bounds

The distance of a linear code can be viewed as a map (and its restriction):

$$\mathrm{d} : \mathcal{L} \longrightarrow \mathbb{N} \cup \{\infty\}\,, \qquad \mathrm{d} : \mathcal{C} \longrightarrow \mathbb{N} \cup \{\infty\}\,,$$

if we adopt the convention that the distance of zero codes is ∞.

Definition 1.3.1. *A map $\delta : \mathcal{L} \to \mathbb{N} \cup \{\infty\}$ is called*

- *a* `lower bound` *on $\mathcal{L}$, if $\delta(C) \leq \mathrm{d}(C)$, $\quad \forall\ C\ \in\ \mathcal{L}$*
- *an* `upper bound` *on $\mathcal{L}$, if $\delta(C) \geq \mathrm{d}(C)$, $\quad \forall\ C\ \in\ \mathcal{L}$.*

  Analogously for a map $\delta : \mathcal{C} \to \mathbb{N} \cup \{\infty\}$.

Definition 1.3.2. *Let C be a code in $\mathcal{C}$ and $\mathcal{F} \subseteq \mathcal{C}$. Let δ be a bound on $\mathcal{C}$ (either lower or upper). We say that:*

- *δ is* `tight` *on C, if $\delta(C) = \mathrm{d}(C)$,*
- *δ is* `tight` *on $\mathcal{F}$, if $\delta(C) = \mathrm{d}(C)$, $\quad \forall\ C \in \mathcal{F}$.*

From now on a “bound” will actually be a lower bound on $\mathcal{C}$.

Definition 1.3.3. *Let C be a code in $\mathcal{C}$ and $\mathcal{F} \subseteq \mathcal{C}$. Let δ_1 and δ_2 be two bounds. We say that:*

- *δ_1 is* `sharper` *than δ_2 on C, if $\delta_1(C) \geq \delta_2(C)$,*
- *δ_1 is* `sharper` *than δ_2 on $\mathcal{F}$, if $\delta_1(C) \geq \delta_2(C)$, $\quad \forall\ C \in \mathcal{F}$.*
- *δ_1 is* `tighter` *than δ_2 on $\mathcal{F}$, if*

$$|\{C \mid C \in \mathcal{F},\ \delta_1 \textit{ is tight on } C\}| \geq |\{C \mid C \in \mathcal{F},\ \delta_2 \textit{ is tight on } C\}|.$$

The last definition we need is the following.

Definition 1.3.4. *Let δ be a bound. We say that δ is* `monotone` *if for any cyclic code C and any cyclic subcode C' of C, we have*

$$\delta(C) \leq \delta(C')\,.$$

Our Tools

Here we present the main tools we use in Chapter 3 and Chapter 4. This instruments are classical in coding theory. Our principal references are [BS07, Sch88, Cha98, MS81] for Section 2.1 and [BS07, Sch88] for Section 2.2.

2.1 DFT and cyclic codes

Let $\mathbb{K}$ be a field. Let α be a primitive n-th root of unity over $\mathbb{K}$.

Let A be any matrix over $\mathbb{K}$. We denote by $\mathrm{rk}(A)$ the rank of A.

Definition 2.1.1. *Let $\bar{a} = (a_1, \dots, a_n)$ be a vector over $\mathbb{K}$. We denote by $\mathbf{M}(\bar{\mathbf{a}})$ the matrix:*

$$M(\bar{a}) = \begin{pmatrix} a_1 & a_2 & \dots & a_{n-1} & a_n \\ a_2 & a_3 & \dots & a_n & a_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n & a_1 & \dots & a_{n-2} & a_{n-1} \end{pmatrix}$$

and we say that $M(\bar{a})$ is the `matrix associated` to $\bar{a}$.

By definition, $M(\bar{a})$ is a circulant matrix (i.e. its rows are obtained from the first one by successive shifts).

Definition 2.1.2. *Let $\bar{a} = (a_0, \dots, a_{n-1})$ be a vector over $\mathbb{F}_q$, α a primitive root in $\mathbb{F}$. Let $i \in \mathbb{Z}$, we define $A_i = \bar{a}(\alpha^i) = \sum_{j=0}^{n-1} a_j \alpha^{ij}$. The `Discrete Fourier Transform` (or `DFT` for short) of $\bar{a}$ is the vector:*

$$\mathrm{DFT}(\bar{a}) = (A_0, \dots, A_{n-1}).$$

The polynomial $A(x) = \sum_{i=1}^{n} A_i x^{n-i} \in \mathbb{F}[x]$ is called the `Mattson-Solomon polynomial` (`MS polynomial` for short) of $\bar{a}$.

Note that $A_0 = A_n$.

We have an useful inversion formula, which allows us to recover $\bar{a}$ from $A(x)$.

Theorem 2.1.3. *[Inversion formula] Let $\bar{a}$, α, $A(x)$ as above. The vector $\bar{a}$ is recovered from $A(x)$ by*

$$a_i = \frac{1}{n}A(\alpha^i), \quad i = 0, \dots, n-1$$

$$\bar{a} = \frac{1}{n}(A(1), A(\alpha), \dots, A(\alpha^{n-1})), \qquad \bar{a}(x) = \frac{1}{n}\sum_{i=0}^{n-1} A(\alpha^i)x^i.$$

Proof. See [MS81]. □

From Theorem 2.1.3 it is possible to deduce that the weight of $\bar{a}$ is n minus the number of zeros of $A(x)$.

Remark 2.1.4. We note that it is possible to define the MS polynomial when $(q, n) \neq 1$ but in general it is not invertible. This is another reason for which we always assume n and q coprime.

Remark 2.1.5. Let $C \in \mathcal{C}_{q,n}$. If we represent a word $c \in C$ as a polynomial in $\mathbb{F}_q[x]$, then the zero components of its DFT correspond to the zeros of c, since $A_i = c(\alpha^i)$ for any i. Moreover, since $c(\alpha^{iq}) = c(\alpha^i)^q$, we also have $A_{(iq)_n} = (A_i)^q$ for any i.

The precise correspondence between codewords and their DFT's is described in the following theorem, which is a rephrasing of classical results.

Theorem 2.1.6. *Let S be a subset of $\{0, \dots, n-1\}$ which is invariant under multiplication by q modulo n. Let L be the subspace of $\mathbb{F}^n$ whose elements are n-tuples $(A_0, \dots A_{n-1})$ satisfying*

$$A_{(qs)_n} = (A_s)^q \quad \textit{and} \quad A_s = 0 \quad \textit{for any } s \in S$$

Let C be the cyclic code of length n over $\mathbb{F}_q$ with defining set S. Then there is a one-to-one correspondence between the codewords of C and the vectors of L, given by $c \mapsto \mathrm{DFT}(c)$.

Given any vector $c = (c_0, \dots, c_{n-1})$ of $(\mathbb{F}_q)^n$, we consider the diagonal matrix D_c and the Vandermonde matrix F, defined as:

$$D_c = \begin{pmatrix} c_0 & 0 & \dots & 0 \\ 0 & c_1 & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & c_{n-1} \end{pmatrix} \qquad F = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \alpha^1 & \dots & \alpha^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-1} & \dots & \alpha^{(n-1)(n-1)} \end{pmatrix}.$$

It is clear that we have $\mathrm{rk}(D_c) = \mathrm{w}(c)$ and $\mathrm{rk}(F) = n$, since the α^j are distinct for $0 \leq j \leq n-1$. Let $\mathrm{DFT}(c) = (C_0, C_1, \dots, C_{n-1})$, we have the following identity due to Blahut ([Bla83]):

$$\begin{pmatrix} C_0 & C_1 & \dots & C_{n-1} \\ C_{n-1} & C_0 & \dots & C_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ C_1 & C_2 & \dots & C_0 \end{pmatrix} = F^{-1} \begin{pmatrix} c_0 & 0 & \dots & 0 \\ 0 & c_1 & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & c_{n-1} \end{pmatrix} F \tag{2.1}$$

If C_1 and C_2 are naturally equivalent cyclic codes of length n, we know a permutation $\sigma \in \mathrm{Sym}(n)$ such that for any $c_1 \in C_1$ there exists $c_2 \in C_2$ with $c_2 = c_1\sigma$ and this permutation acts as $i\sigma = (is)_n$ for some $s \in \{0, \dots, n-1\}$, $(s, n) = 1$. We have that $D_{c_2} = P_\sigma^T D_{c_1} P_\sigma$. From this fact, we are able to prove that $M(\mathrm{DFT}(c_1))$ and $M(\mathrm{DFT}(c_2))$ are closely related. We claim the following proposition.

Proposition 2.1.7. *Let C_1 and C_2 be two naturally equivalent cyclic codes. Then there is a permutation matrix, P_λ, such that for any $c_2 \in C_2$ there is an unique $c_1 \in C_1$ s.t.*

$$M(\mathrm{DFT}(c_2)) = P_\lambda M(\mathrm{DFT}(c_1)) P_\lambda^T$$

Proof. Let n be the length of C_1 and C_2. Since C_1 and C_2 are naturally equivalent, there is a permutation σ of the form $i\sigma = (is)_n$ with $(s, n) = 1$, such that $C_2 = C_1 P_\sigma$. Let $c_2 = c_1\sigma$, $c_1 \in C_1$, $c_2 \in C_2$. Then we have

$$M(\mathrm{DFT}(c_2)) = F^{-1} D_{c_2} F = F^{-1} P_\sigma^T D_{c_1} P_\sigma F.$$

We claim that the matrix $F^{-1} P_\sigma^T F$ is a permutation matrix. If our claim is true, we set $P_\lambda = F^{-1} P_\sigma^T F$, obtaining:

$$\begin{aligned} P_\lambda M(\mathrm{DFT}(c_1)) P_\lambda^T &= P_\lambda \ F^{-1} D_{c_1} F \ P_\lambda^T \\ &= P_\lambda \ F^{-1} D_{c_1} F \ P_\lambda^{-1} \\ &= (F^{-1} P_\sigma^T F) \ F^{-1} D_{c_1} F \ (F^{-1} P_\sigma^T F)^{-1} \\ &= (F^{-1} P_\sigma^T) D_{c_1} (P_\sigma F) \\ &= F^{-1} D_{c_2} F \\ &= M(\mathrm{DFT}(c_2)) \end{aligned}$$

Since F is a Vandermonde matrix, we have that (see [AL69]) its inverse is

$$F^{-1} = \frac{1}{n} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \alpha^{-1} & \dots & \alpha^{-(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{-(n-1)} & \dots & \alpha^{-(n-1)(n-1)} \end{pmatrix}.$$

while the permutation matrix corresponds to the permutation σ^{-1} which is of the form $i\sigma^{-1} = (it)_n$ with $(t, n) = 1$, because it must hold $st = 1 \mod n$. Thus we have $F^{-1}P_\sigma^T = F^{-1}P_{\sigma^{-1}}$ and:

$$\begin{aligned}
F^{-1}P_{\sigma^{-1}} &= \frac{1}{n}\begin{pmatrix} (\alpha^0)^0 & (\alpha^0)^1 & \dots & (\alpha^0)^{(n-1)} \\ (\alpha^{-1})^0 & (\alpha^{-1})^1 & \dots & (\alpha^{-1})^{(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha^{-(n-1)})^0 & (\alpha^{-(n-1)})^1 & \dots & (\alpha^{-(n-1)})^{(n-1)} \end{pmatrix} P_{\sigma^{-1}} \\
&= \frac{1}{n}\begin{pmatrix} (\alpha^0)^{0\sigma^{-1}} & (\alpha^0)^{1\sigma^{-1}} & \dots & (\alpha^0)^{(n-1)\sigma^{-1}} \\ (\alpha^{-1})^{0\sigma^{-1}} & (\alpha^{-1})^{1\sigma^{-1}} & \dots & (\alpha^{-1})^{(n-1)\sigma^{-1}} \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha^{-(n-1)})^{0\sigma^{-1}} & (\alpha^{-(n-1)})^{1\sigma^{-1}} & \dots & (\alpha^{-(n-1)})^{(n-1)\sigma^{-1}} \end{pmatrix} \\
&= \frac{1}{n}\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & (\alpha^{-1})^t & \dots & (\alpha^{-1})^{(n-1)\cdot t} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (\alpha^{-(n-1)})^t & \dots & (\alpha^{-(n-1)})^{(n-1)\cdot t} \end{pmatrix} \\
= \frac{1}{n}T.
\end{aligned}$$

We compute the product $L = TF$. We note that the i-th row of T is $\left(1\ (\alpha^{-t})^{i-1}\ \dots\ (\alpha^{-t})^{(i-1)(n-1)}\right)$, while the j-th column of F is $\left(1\ \alpha^{j-1}\ \dots\ \alpha^{(j-1)(n-1)}\right)^T$. Thus we have:

$$l_{ij} = \sum_{k=0}^{n-1} \alpha^{k(j-1-t(i-1))} = \begin{cases} n & \text{if } (j-1) - t(i-1) = 0 \mod n, \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where we have used the classical result $\sum_{\beta^n=1} \beta = 0$, which holds for any field $\mathbb{F}_q$ and for any $n \geq 2$, provided $(n, q) = 1$. From (2.2) we have that $l_{ij} = n$ if and only if $j = t(i-1) + 1 \mod n$, which means $j = (t(i-1))_n + 1$, since $1 \leq j \leq n$. Let us consider any row of $P_\lambda = \frac{1}{n}L$, say the i-th. We have proved that its j-th entry is 1 if $j = (t(i-1))_n + 1$, which happens only one per rows, and it is zero otherwise. Then it is sufficient to note that, by definition, P_λ is an invertible square matrix, to conclude that it is a permutation matrix. □

We collect in one statement some results from [Bla83], [Sch88] and [MS88] (mainly the "Zero-Location Theorem" in [MS88]), which follows immediately from (2.1)

Theorem 2.1.8. *Let C be a cyclic code and let* $\mathrm{DFT}(C)$ *be the code formed by the Discrete Fourier Transforms of the words of C. Then the distance of C is the* *minimum of the ranks of the matrices associated to all nonzero words in* $\mathrm{DFT}(C)$*, i.e.*

$$\mathrm{d}(C) = \min\{\mathrm{rk}(M(\mathrm{DFT}(c))) \mid c \in C, c \neq 0\}.$$

Thus, the problem of finding the distance of a code is equivalent to finding the minimum rank of the corresponding set of matrices. In particular, any bound for one is also a bound for the other one.

2.2 The set $\mathcal{U}$

We present some notation from [BS06].

Definition 2.2.1. *Let $\mathcal{U}$ be a set formed by three elements, which we call $\{\Delta, \Delta^{\!\!+}, 0\}$. We endow $\mathcal{U}$ with two operations, sum and product, according to the following logical tables:*

$\cdot$	Δ	$\Delta^{\!\!+}$	0
Δ	Δ	Δ	0
$\Delta^{\!\!+}$	Δ	$\Delta^{\!\!+}$	0
0	0	0	0

$+$	Δ	$\Delta^{\!\!+}$	0
Δ	Δ	Δ	Δ
$\Delta^{\!\!+}$	Δ	Δ	$\Delta^{\!\!+}$
0	Δ	$\Delta^{\!\!+}$	0

Table 2.1: Multiplication and sum in $\mathcal{U}$

The set $\mathcal{U}$ plays the role of a field where we have partial information on the element values. More precisely, let $\mathbb{K}$ be any field, we say that:

- $\Delta^{\!\!+}$ represents an element of $\mathbb{K}$ for which we know it is different from zero,
- 0 represents an element of $\mathbb{K}$ for which we know it is zero,
- Δ represents an element of $\mathbb{K}$ for which we do not know if it is zero or we do not care.

One should regard an element of $\mathcal{U}$ as *the information we have* on a field element, rather that a way to indicate its value.

Example 2.2.2. Sum and product are defined over $\mathcal{U}$ following the interpretation of the symbols 0, Δ, $\Delta^{\!\!+}$. In fact, $\Delta^{\!\!+} \cdot \Delta^{\!\!+} = \Delta^{\!\!+}$ is equivalent to saying that the product of two non-zero elements is different from zero, while $\Delta^{\!\!+} + \Delta^{\!\!+} = \Delta$ is equivalent to saying that the sum of two non-zero elements could be zero or non-zero.

Although $\mathcal{U}$ is not a field and $\mathcal{U}^n$ is not a vector space, it is convenient to use some terminology traditionally associated to vector spaces, paying attention to define rigorously our notation.

Definition 2.2.3. *Let $u = (u_0, \ldots, u_{n-1})$ be any element of $\mathcal{U}^n$. We say that u is a* `vector`. *We also write $u[i] = u_{i-1}$ for any $1 \leq i \leq n$.*

Remark 2.2.4. Let $k \in \mathbb{Z}$ be any integer and $\mathbf{u} \in \mathcal{U}^n$. For convenience, sometimes we write $\mathbf{u}[k]$, meaning:

$$\mathbf{u}[k] = \begin{cases} \mathbf{u}[(k)_n] & \text{if } (k)_n \neq 0 \\ \mathbf{u}[n] & \text{otherwise.} \end{cases}$$

Definition 2.2.5. *Let $n \geq 1$ and $S \subseteq \{0, \ldots, n-1\}$.*
We denote by $R(n, S)$ the vector $(u_0, \ldots, u_{n-1})$ in $\mathcal{U}^n$ such that $u_i = 0$, if i is in S, $u_i = \Delta$ otherwise.
We denote by $\hat{R}(n, S)$ the vector $(u_0, \ldots, u_{n-1})$ in $\mathcal{U}^n$ such that $u_i = 0$, if i is in S, $u_i = \Delta^{+}$ otherwise.

Note that, if $C \in \mathcal{C}_{q,n}$ and α is a primitive n-th root of unity over $\mathbb{F}_q$, then $R(n, S_{C,\alpha})$ and $\hat{R}(n, S_{C,\alpha})$ are well-defined vectors in $\mathcal{U}^n$.

Definition 2.2.6. *Let $\mathbf{v} \in \mathcal{U}^n$. We denote by $M(\mathbf{v}) \in \mathcal{U}^{n \times n}$ the circulant matrix obtained from vector $\mathbf{v}$, i.e. the matrix whose first row is $\mathbf{v}$ and whose other rows are obtained by cyclic shifting.*

We want to introduce the notion of linear dependence in $\mathcal{U}^n$. We want that a set of vectors is linear independent in $\mathcal{U}^n$ if they correspond to a set of linear independent vectors in every vector space $\mathbb{K}^n$. To define this notion in a rigorous way, we need a couple of definitions, presented here for the first time.

Definition 2.2.7. *Let $n \geq 1$ be a natural number, $\mathbf{u} = (u_0, \ldots, u_{n-1}) \in \mathcal{U}^n$. Let $\mathbb{K}$ be any field. An* `instance` *of $\mathbf{u}$ over $\mathbb{K}$ is any vector $v = (v_0, \ldots, v_{n-1}) \in \mathbb{K}^n$ such that for $0 \leq i \leq n-1$:*

1. *$v_i = 0$ if $u_i = 0$,*
2. *$v_i \neq 0$ if $u_i = \Delta^{+}$.*

The set of all instances of $\mathbf{u}$ over $\mathbb{K}$ is called the `instantiation` *of $\mathbf{u}$ over $\mathbb{K}$ and we write* $\mathtt{In}(\mathbf{u}, \mathbb{K}) = \{\, v \in \mathbb{K}^n \mid v \text{ is an instance of } \mathbf{u} \text{ over } \mathbb{K} \,\}$.

Remark 2.2.8. Note that in Definition 2.2.7 we did not specify the value of v_i when $u_i = \Delta$, so v_i can be freely chosen for this value of i.

Example 2.2.9. Let us consider $\mathbb{K} = \mathbb{F}_2$.

- if $\mathbf{u} = (0, \Delta, \Delta^{+}) \in \mathcal{U}^3$, then $\mathtt{In}(\mathbf{u}, \mathbb{F}_2) = \{\, (0,0,1), (0,1,1) \,\}$
- if $\mathbf{u} = (0, \Delta^{+}, \Delta) \in \mathcal{U}^3$, then $\mathtt{In}(\mathbf{u}, \mathbb{F}_2) = \{\, (0,1,0), (0,1,1) \,\}$.

Definition 2.2.10. *Let $s \geq 1$. We say that $\mathbf{u}^1, \dots, \mathbf{u}^s \in \mathcal{U}^n$ are **linear independent** if for any field $\mathbb{K}$, for any $v^i \in \mathtt{In}(\mathbf{u}^i, \mathbb{K})$ with $1 \leq i \leq s$, we have that $\{\, v^i \,\}_{1 \leq i \leq s}$ are linear independent (over $\mathbb{K}$).*

In other words, for any instance set $\{\, v^1, \dots, v^s \,\}$, for any $\{\, \lambda_i \,\}_{1 \leq i \leq s} \subseteq \mathbb{K}$:

$$\sum_{i=1}^{s} \lambda_i v^i = 0 \iff \lambda_1 = \cdots = \lambda_s = 0.$$

In a similar way we can also define the instance of a matrix.

Definition 2.2.11. *Let $A \in \mathcal{U}^{m \times n}$, $A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$. Let $\mathbb{K}$ be any field. An **instance** of A over $\mathbb{K}$ is any matrix $B \in \mathbb{K}^{m \times n}$, $B = (b_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$, such that:*

1. *$b_{ij} = 0$ if $a_{ij} = 0$,*
2. *$b_{ij} \neq 0$ if $a_{ij} = \Delta^{+}$.*

*The set of all instances of A is called **instantiation** of A over $\mathbb{K}$ and we write $\mathtt{In}(A, \mathbb{K}) = \{\, B \in \mathbb{K}^{m \times n} \mid B$ is an instance of $A \,\}$.*

Definition 2.2.12. *Given a matrix A over $\mathcal{U}$, we denote by $\mathrm{rk}(A)$ the **rank** of A, i.e. the largest r s.t. there exists a set of r linear independent rows.*

From Definition 2.2.12, the following fact is straightforward.

Fact 2.2.13. *Let $A \in \mathcal{U}^{m \times n}$ be any matrix over $\mathcal{U}$. Then*

$$\mathrm{rk}(A) = \min_{\mathbb{K}} \{\, \mathrm{rk}(B) \mid B \in \mathtt{In}(A, \mathbb{K})) \,\}\,.$$

It is easy to see that this notion of rank for rows is equivalent to a notion of ranks for columns, since this equivalence holds over any field.

Our interest in ranks over $\mathcal{U}$ lies in the following theorem.

Proposition 2.2.14. *Let $M = (m_{i,j})$ be an $r \times s$ matrix over a field $\mathbb{K}$. Let $\hat{M} = (\hat{m}_{i,j})$ be the $r \times s$ matrix over $\mathcal{U}$ s.t. $\hat{m}_{i,j} = 0$ if $m_{i,j} = 0$ and $\hat{m}_{i,j} = \Delta^{+}$ otherwise. Then*

$$\mathrm{rk}(\hat{M}) \leq \mathrm{rk}(M)\ .$$

Proof. By construction of $\hat{M}$, we have $M \in \mathtt{In}(\hat{M}, \mathbb{K})$. Then thanks to Fact 2.2.13 our claim follows. □

Definition 2.2.15. *Given a vector* $\mathbf{v} \in \mathcal{U}^n$ *we denote by* $\mathcal{A}(\mathbf{v})$ *the set of vectors* $\mathbf{u} \in \mathcal{U}^n \setminus \mathbf{0}$ *s.t.*

- $\mathbf{u}[i] = 0$, *if* $\mathbf{v}[i] = 0$,
- $\mathbf{u}[i] = \Delta\!\!^{+}$, *if* $\mathbf{v}[i] = \Delta\!\!^{+}$,
- $\mathbf{u}[i] = \Delta\!\!^{+}$, *or* $\mathbf{u}[i] = 0$ *if* $\mathbf{v}[i] = \Delta$.

Observe that if there is at least one component of $\mathbf{v}$ equal to $\Delta\!\!^{+}$ then $|\mathcal{A}(\mathbf{v})| = 2^s$, where s represents the number of components of $\mathbf{v}$ equal to Δ. Otherwise $|\mathcal{A}(\mathbf{v})| = 2^s - 1$.

Theorem 2.2.16. *Let* C *be a cyclic code of length* n*, defining set* $S_{C,\alpha}$ *and distance* d*. Then:*

$$\min\{\mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n, S_{C,\alpha}))\} \leq d\,.$$

Proof. From Proposition 2.2.14 and Theorem 2.1.8. □

Root bounds

This chapter belongs to a work joint with E. Betti, relates the results contained in the unpublished paper [BS07] but also advances significantly on [BS07], especially in Theorem 3.2.18, Proposition 3.3.3 (this was claimed in [BS07] but without a convincing proof) and Theorem 3.5.8.

3.1 General settings

In this chapter we propose a family of bounds and study their properties.

We need a definition to fix our setting.

Definition 3.1.1. *We denote by $\mathcal{D}$ the following subset of $\mathbb{N} \times 2^{\mathbb{N}}$:*

$$(n, S) \in \mathcal{D} \qquad \Longleftrightarrow \qquad n \geq 1 \,,\; S \subseteq \{0, \dots, n-1\} \,.$$

Let $(n, S) \in \mathcal{D}$. Let $S = \{i_1, \dots, i_m\}$. We denote by $(n, S)^{\#}$ the following set of subsets of $\{0, \dots, n-1\}$

$$(n, S)^{\#} \;=\; \{S_1, \dots, S_r\} \,,$$

where $r = |\mathbb{Z}_n^*|$ and for any $l \in \mathbb{Z}_n^*$ there is one and only one j such that $S_j = \{(li_h)_n \mid 1 \leq h \leq m\}$.

Note that $S \in (n, S)^{\#}$ and $|S_h| = |S|$ for any h.

Note that in Definition 3.1.1 we do not require that $S \neq S_{\bar{l}}$ for $l \neq \bar{l}$. The two extreme cases are given by $S = \{1\}$, where $S_l \neq S_{\bar{l}}$ for any $l \neq \bar{l}$, and by $S = \{0\}$, where $S_l = S_{\bar{l}}$ for any $\bar{l}$.

Definition 3.1.2. *We denote by χ the map $\chi : \mathcal{C} \mapsto \mathbb{N}$ s.t. $\chi(C) = p$, if C is over $\mathbb{F}_q$ and p is the characteristic of $\mathbb{F}_q$.*

Using a function $\zeta \in \mathcal{Z}$ (Definition 1.2.15), we define a map from $\mathcal{C}$ to $\mathcal{D}$:

$$\phi_\zeta : \mathcal{C} \to \mathcal{D}, \quad \phi_\zeta(C) \;=\; (n, S_{C,\alpha}) \,, \tag{3.1}$$

where $\alpha = \zeta(\chi(C), n)$.

Proposition 3.1.3. *For any $\zeta \in \mathcal{Z}$, map ϕ_ζ is surjective.*

Proof. Given a pair $(n, \{i_1, \ldots, i_m\})$ in $\mathcal{D}$, take any prime p. Let $\alpha = \zeta(p, n)$ and let $\mathbb{F}_q \subseteq \overline{\mathbb{F}}_p$ be a finite field containing α. Let C be the cyclic code over $\mathbb{F}_q$ generated by $g = (x - \alpha^{i_1})(x - \alpha^{i_2}) \cdots (x - \alpha^{i_m})$ and with length n.
Clearly, $\phi_\zeta(C) = (n, \{i_1, \ldots, i_m\})$. □

Definition 3.1.4. *A* `root function` *is a map $f : \mathcal{D} \to \mathbb{N} \cup \{\infty\}$ such that:*

$$\forall \zeta \in \mathcal{Z}, \forall\ C \in \mathcal{C}, \quad f \circ \phi_\zeta(C) \leq \mathrm{d}(C)\,. \tag{3.2}$$

We denote by $\mathcal{R}$ the class of all root functions.

Given $f \in \mathcal{R}$, we say that f is `invariant` *if $f(n, S) = f(n, T)$, for any $T \in (n, S)^\#$. We also denote by $f^\#$ the map $f^\#(n, S) = \max_{T \in (n,S)^\#} f(n, T)$.*

For any $\zeta \in \mathcal{Z}$ and any $f \in \mathcal{R}$, the composite map $f_{\mathcal{D},\zeta} = f \circ \phi_\zeta : \mathcal{C} \mapsto \mathbb{N} \cup \{\infty\}$ is called the `root bound` *associated to f and ζ. If f is invariant, we say that $f_{\mathcal{D},\zeta}$ is invariant. We denote by $\mathcal{R}_D$ the class of all root bounds.*

Due to (3.2), root bounds are actually lower bounds for the distance on $\mathcal{C}$.

If $f \in \mathcal{R}$ is invariant, we have that $f_{\mathcal{D},\zeta} = f_{\mathcal{D},\zeta'}$ for any ζ and ζ', and so we just write $f_{\mathcal{D}}$.

Given any $f \in \mathcal{R}$, $f^\#$ represents the "invariant version" of f, as explained in the next proposition.

Proposition 3.1.5. *For any $f \in \mathcal{R}$, we have:*

1. *$f^\# \in \mathcal{R}$,*
2. *$f^\#$ is invariant,*
3. *$f \leq f^\#$,*
4. *$f^\#_{\mathcal{D}} = \max_{\zeta \in \mathcal{Z}} f_{\mathcal{D},\zeta}$.*

Proof.

1) Let $C \in \mathcal{C}$. Then $C \in \mathcal{C}_{q,n}$ for some q and n, and $p = \chi(C)$. Let $\zeta \in \mathcal{Z}$ and $\alpha = \zeta(p, n)$. We have to prove that $(f^\# \circ \phi_\zeta)(C) \leq \mathrm{d}(C)$. Let $(n, S_{C,\alpha}) = \phi_\zeta(C)$. From the definition of $f^\#$ we have that $f^\#(n, S_{C,\alpha}) = f(n, T)$, for some $T \in (n, S_{C,\alpha})^\#$. From Theorem 1.2.21, there is a code D naturally equivalent to C such that $S_{D,\alpha} = T$. Theorem 1.2.20 guarantees that $\mathrm{d}(C) = \mathrm{d}(D)$, so we have:

$$(f^\# \circ \phi_\zeta)(C) = f(n, S_{D,\alpha}) = (f \circ \phi_\zeta)(D) \leq \mathrm{d}(D) = \mathrm{d}(C).$$

2) Let $(n, S) \in \mathcal{D}$ then for any $T \in (n, S)^\#$ we have $(n, S)^\# = (n, T)^\#$. So:

$$f^\#(n, S) = \max_{H \in (n,S)^\#} f(n, H) = \max_{H \in (n,T)^\#} f(n, H) = f^\#(n, T).$$

3) Recalling that for any $(n, S) \in \mathcal{D}$, $S \in (n, S)^{\#}$, we conclude that:

$$f^{\#}(n, S) = \max_{T \in (n,S)^{\#}} f(n, T) \geq f(n, S).$$

4) Let $C \in \mathcal{C}$. Then $C \in \mathcal{C}_{q,n}$ for some q and n, and $p = \chi(C)$. Let $\zeta \in \mathcal{Z}$ and $\alpha = \zeta(p, n)$. We have to prove that $f^{\#}_{\mathcal{D}}(C) = (f^{\#} \circ \phi_\zeta)(C) = \max_{\zeta \in \mathcal{Z}}(f \circ \phi_\zeta)(C)$. Since $(n, S_{C,\alpha_\zeta}) = \phi_\zeta(C)$ it holds $(f^{\#} \circ \phi_\zeta)(C) = f^{\#}(n, S_{C,\alpha_\zeta}) = \max_{T \in (n, S_{C,\alpha_\zeta})^{\#}} f(n, T)$. Thanks to Theorem 1.2.21, for any $T \in (n, S_{C,\alpha_\zeta})^{\#}$ there is a code D naturally equivalent to C (we write $D \sim C$) such that $T = S_{D,\alpha_\zeta}$. So we have:

$$f^{\#}_{\mathcal{D}}(C) = \max_{T \in (n, S_{C,\alpha_\zeta})^{\#}} f(n, T) = \max_{D \sim C} f(n, S_{D,\alpha_\zeta}).$$

From Definition 1.2.17 and Theorem 1.2.20 we have $\max_{D \sim C} f(n, S_{D,\alpha_\zeta}) = \max_{\zeta' \in \mathcal{Z}} f(n, S_{C,\alpha_{\zeta'}})$, thus:

$$f^{\#}_{\mathcal{D}}(C) = \max_{\zeta' \in \mathcal{Z}} f(n, S_{C,\alpha_{\zeta'}}) = \max_{\zeta' \in \mathcal{Z}} (f \circ \phi_{\zeta'})(C) = \max_{\zeta' \in \mathcal{Z}} f_{\mathcal{D},\zeta'}(C).$$

□

The following remark is essential to understand our approach in this chapter and actually in most of this thesis.

Remark 3.1.6. A lower bound (see Definition 1.3.1) is a map that gives an estimate on the minimum distance of a cyclic code. With a root bound this estimate is given while ignoring all information about the code, except the length and a defining set. In particular, no information on the underlying field is used.

$$\begin{tikzcd}
\mathcal{C} \arrow[rr, "f \circ \phi_\zeta"] \arrow[dr, "\phi_\zeta"'] & & \mathbb{N} \\
& \mathcal{D} \arrow[ur, "f"'] &
\end{tikzcd}$$

We can rewrite a known theorem on sub-field subcodes of cyclic codes, using our notation.

Theorem 3.1.7 ([MS88]). *Let $C_1 \in \mathcal{C}_{q_1,n}$, $C_2 \in \mathcal{C}_{q_2,n}$. Let $\zeta \in \mathcal{Z}$. We have*

$$\mathbb{F}_{q_1} \subseteq \mathbb{F}_{q_2},\ \phi_\zeta(C_1) = \phi_\zeta(C_2) \quad \Longrightarrow \quad \mathrm{d}(C_1) = \mathrm{d}(C_2)\,.$$

From this theorem we can easily get a slightly more general statement.

Proposition 3.1.8. *Let $C_1 \in \mathcal{C}_{q_1,n}$, $C_2 \in \mathcal{C}_{q_2,n}$. Let $\zeta \in \mathcal{Z}$. We have*

$$\chi(C_1) = \chi(C_2),\, \phi_\zeta(C_1) = \phi_\zeta(C_2) \quad \Longrightarrow \quad \mathrm{d}(C_1) = \mathrm{d}(C_2)\,.$$

Proof. Let $p = \chi(C_1) = \chi(C_2)$. Then $q_1 = p^{r_1}$ and $q_2 = p^{r_2}$, for some $r_1, r_2 \geq 1$. Let $Q = p^{r_1 r_2}$. We have $\mathbb{F}_{q_1}, \mathbb{F}_{q_2} \subseteq \mathbb{F}_Q$. Consider $C_3 \in \mathcal{C}_{Q,n}$ s.t. $\phi_\zeta(C_3) = \phi_\zeta(C_1)$. By Theorem 3.1.7 we have $\mathrm{d}(C_3) = \mathrm{d}(C_1)$ (because $\mathbb{F}_{q_1} \subseteq \mathbb{F}_Q$) and $\mathrm{d}(C_3) = \mathrm{d}(C_2)$ (because $\mathbb{F}_{q_2} \subseteq \mathbb{F}_Q$). □

In other words, a defining set, a length and a field characteristic uniquely determine a distance.

Definition 3.1.9. *Let f be a root function. We say that f is* `monotone` *if for any (n, S) and (n', S') in $\mathcal{D}$ we have*

$$n = n',\, S \subseteq S' \quad \Longrightarrow \quad f(n, S) \leq f(n, S')$$

Any root bound associated to f is called a `monotone root bound`.

Thanks to the next result (Theorem 3.1.10) the two terms "a monotone root bound" and "a monotone bound which is a root bound" correspond to the same notion.

Theorem 3.1.10. *Let $\delta \in \mathcal{R}_D$ be a monotone root bound. Let C be a cyclic code and C' be a cyclic subcode of C. Then*

$$\delta(C) \leq \delta(C')$$

Proof. We have that $\delta = f \circ \phi_\zeta$, for a root function f and a map $\zeta \in \mathcal{Z}$. Since C' is a cyclic subcode of C, we have that $\phi_\zeta(C) = (n, S)$ and $\phi_\zeta(C') = (n, S')$, with $S' \supset S$. By definition of monotone root bound we have that $f(n, S) \leq f(n, S')$ and then $\delta(C) \leq \delta(C')$. □

For any $f \in \mathcal{R}$, we denote by f^* the map

$$f^*(n, S) = \max\left\{ f(n, S') \mid S' \subseteq S \right\}. \tag{3.3}$$

The f^* construction is useful, since it produces the *least* monotone root function from f, as detailed in next proposition.

Proposition 3.1.11. *Let $f \in \mathcal{R}$. We have:*

1. *f^* is a root function,*

2. *f^* is monotone,*

3. $f \leq f^*$,

4. if g *is any monotone root functions s.t.* $f \leq g$*, then* $f^* \leq g$.

Proof.

1) Let $C \in \mathcal{C}_{q,n}$ and $\zeta \in \mathcal{Z}$. We have to prove that $(f^* \circ \phi_\zeta)(C) \leq \mathrm{d}(C)$.
Let $\mathbb{F} = \mathbb{F}_Q$ be the splitting field of $x^n - 1$ over $\mathbb{F}_q$. Let us consider $\tilde{C} \in \mathcal{C}_{Q,n}$ such that $\phi_\zeta(\tilde{C}) = \phi_\zeta(C)$. We have $\mathrm{d}(C) = \mathrm{d}(\tilde{C})$ by Theorem 3.1.7 and so it is enough to prove $(f^* \circ \phi_\zeta)(\tilde{C}) \leq \mathrm{d}(\tilde{C})$. Let $(n, S) = \phi_\zeta(\tilde{C})$. By definition of f^* we have $f^*(n, S) = f(n, S')$, for some $S' \subseteq S$. Let $C' \in \mathcal{C}_{Q,n}$ s.t. $\phi_\zeta(C') = (n, S')$. We have that $\tilde{C} < C'$ and hence $\mathrm{d}(\tilde{C}) \geq \mathrm{d}(C')$. Putting all together, we get

$$(f^* \circ \phi_\zeta)(C) = f^*(n, S) = f(n, S') \leq \mathrm{d}(C') \leq \mathrm{d}(\tilde{C}) = \mathrm{d}(C).$$

2) If $S \subseteq T$ then $\{S' \mid S' \subseteq S\} \subseteq \{T' \mid T' \subseteq T\}$ and hence

$$\max_{T' \subseteq T} f(n, T') \geq \max_{S' \subseteq S} f(n, S').$$

3) It is straightforward, since $S \subseteq S$.

4) Let $(n, S) \in \mathcal{D}$. For any $S' \subseteq S$, $g(n, S) \geq g(n, S') \geq f(n, S')$, so

$$g(n, S) \geq \max_{S' \subseteq S} f(n, S') \ = \ f^*(n, S).$$

□

Clearly, the previous construction can be extended to the corresponding root bounds, but we find it unnecessary to give an explicit statement.

We define a map f from $\mathcal{D}$ to $\mathbb{N} \cup \{\infty\}$, as follows

$$\mathsf{f}(n, S) = \max\{f(n, S) \mid f \in \mathcal{R}\}\,. \tag{3.4}$$

Theorem 3.1.12. *Map* f *is a root function, which is maximal in* $\mathcal{R}$*, monotone and invariant.*

Proof. Map f is in $\mathcal{R}$, if for any $C \in \mathcal{C}$ and any $\zeta \in \mathcal{Z}$, $\mathsf{f} \circ \phi_\zeta \leq \mathrm{d}(C)$. Let $(n, S) = \phi_\zeta(C)$. There must be an $f \in \mathcal{R}$ s.t. $\mathsf{f}(n, S) = f(n, S)$ (by definition of f) and hence $\mathsf{f} \circ \phi_\zeta = \mathsf{f}(n, S) = f(n, S) = f \circ \phi_\zeta(C) \leq \mathrm{d}(C)$.

It is obvious that f is maximal in $\mathcal{R}$, since for any $(n, S) \in \mathcal{D}$ and any $f \in \mathcal{R}$ we have $\mathsf{f}(n, S) \geq f(n, S)$.

To show that f is monotone, we consider f^*. Then f^* is a monotone root function s.t. $\mathsf{f}^* \geq \mathsf{f}$ (Proposition 3.1.11). By maximality of f we have $\mathsf{f}^* \leq \mathsf{f}$ and hence $\mathsf{f}^* = \mathsf{f}$.

To show that f is invariant, we consider $\mathsf{f}^\#$ and with the same argument as before we obtain that $\mathsf{f} = \mathsf{f}^\#$ (Proposition 3.1.5). □

We can use f to obtain the maximal root bound.

Theorem 3.1.13. *Map $\mathsf{f}_{\mathcal{D}}$ is a monotone invariant root bound, which is maximal in $\mathcal{R}_{\mathcal{D}}$.*

Proof. It follows immediately from Theorem 3.1.12. □

We want to get an alternative characterization for the maximal root bound. We will need a few definitions and lemmas.

Definition 3.1.14. *For any $\zeta \in \mathcal{Z}$ and any $(n, S) \in \mathcal{D}$, we define two sets, $V^{\zeta}_{(n,S)} \subseteq \mathcal{C}$ and $T^{\zeta}_{(n,S)} \subseteq \mathbb{N}$, as follows,*

$$V^{\zeta}_{(n,S)} = \{C \mid C \in \mathcal{C}, \phi_{\zeta}(C) = (n, S)\}$$

$$T^{\zeta}_{(n,S)} = \{\mathrm{d}(C) \mid C \in \mathcal{C}, \phi_{\zeta}(C) = (n, S)\} = \{\mathrm{d}(C)\}_{C \in V^{\zeta}_{(n,S)}}$$

Observe that $V^{\zeta}_{(n,S)} \neq \emptyset$ for any $\zeta \in \mathcal{Z}$ and any $(n, S) \in \mathcal{D}$ (Proposition 3.1.3).

Lemma 3.1.15. *For any $\zeta, \zeta' \in \mathcal{Z}$ and any $(n, S) \in \mathcal{D}$,*

$$|V^{\zeta}_{(n,S)}| = |V^{\zeta'}_{(n,S)}|, \qquad T^{\zeta}_{(n,S)} = T^{\zeta'}_{(n,S)} .$$

Proof. It is enough to construct two maps $\iota_{\zeta,\zeta'}$ and $\iota_{\zeta',\zeta}$, $\iota_{\zeta,\zeta'} : V^{\zeta}_{(n,S)} \mapsto V^{\zeta'}_{(n,S)}$ and $\iota_{\zeta',\zeta} : V^{\zeta'}_{(n,S)} \mapsto V^{\zeta}_{(n,S)}$, s.t.

$$\iota_{\zeta,\zeta'} \circ \iota_{\zeta',\zeta} = \mathrm{id}_{V^{\zeta}_{(n,S)}}, \qquad \iota_{\zeta',\zeta} \circ \iota_{\zeta,\zeta'} = \mathrm{id}_{V^{\zeta'}_{(n,S)}}, \tag{3.5}$$

and

$$\mathrm{d}(C) = \mathrm{d}(\iota_{\zeta',\zeta}(C)),\ \forall C \in V^{\zeta}_{(n,S)}, \qquad \mathrm{d}(C) = \mathrm{d}(\iota_{\zeta,\zeta'}(C)),\ \forall C \in V^{\zeta'}_{(n,S)} . \tag{3.6}$$

Let $C \in V^{\zeta}_{(n,S)}$. Then $C \in \mathcal{C}_{q,n}$, where $\mathbb{F}_q$ is a finite field. By Theorem 1.2.20, there is a unique code $C' \in \mathcal{C}_{q,n}$ s.t. $\phi_{\zeta'}(C') = (n, S)$. By the same theorem, C and C' are naturally equivalent and hence $\mathrm{d}(C) = \mathrm{d}(C')$. But then if we define

$$\iota_{\zeta,\zeta'}(C) = C', \quad \iota_{\zeta',\zeta}(C') = C ,$$

conditions (3.5) and (3.6) are trivially satisfied. □

Definition 3.1.16. *We define a map $\mathsf{g} : \mathcal{D} \mapsto \mathbb{N} \cup \{\infty\}$ by choosing an arbitrary $\zeta \in \mathcal{Z}$ and setting*

$$\mathsf{g}(n, S) = \min T^{\zeta}_{(n,S)}.$$

By Lemma 3.1.15 g does not depend on the particular ζ and hence g is well-defined.

Lemma 3.1.17.

$$\mathsf{g} \in \mathcal{R}\,.$$

Proof. Let $\bar{C} \in \mathcal{C}$ and $\zeta \in \mathcal{Z}$. We have to show that $\mathsf{g} \circ \phi_\zeta(\bar{C}) \leq \mathrm{d}(\bar{C})$.
Let $(n, S) = \phi_\zeta(\bar{C})$. We have $\mathsf{g}(n, S) = \min T^\zeta_{(n,S)}$. But $\bar{C} \in V^\zeta_{(n,S)}$ and so $\mathrm{d}(\bar{C}) \in T^\zeta_{(n,S)}$, which means $\mathsf{g} \circ \phi_\zeta(\bar{C}) = \mathsf{g}(n, S) \leq \mathrm{d}(\bar{C})$. □

We are finally ready for an alternative description of the maximal root bound. We recall that f is the maximal root function.

Theorem 3.1.18.

$$\mathsf{g} = \mathsf{f}\,.$$

Proof. Since f is maximal in $\mathcal{R}$ and $\mathsf{g} \in \mathcal{R}$ (Lemma 3.1.17), we have $\mathsf{g} \leq \mathsf{f}$.

To show $\mathsf{g} \geq \mathsf{f}$ we argument by contradiction, by assuming that there is an $(n, S) \in \mathcal{D}$ such that $\mathsf{g}(n, S) < \mathsf{f}(n, S)$. Let $\zeta \in \mathcal{Z}$. We consider $\bar{C} \in V^\zeta_{(n,S)}$ such that $\mathrm{d}(\bar{C}) = \min T^\zeta_{(n,S)}$ (Proposition 3.1.3). Thus we get a contradiction:

$$\mathrm{d}(\bar{C}) = \mathsf{g} \circ \phi_\zeta(\bar{C}) < \mathsf{f} \circ \phi_\zeta(\bar{C}) \quad \text{and} \quad \mathsf{f} \in \mathcal{R}\,.$$

□

Corollary 3.1.19. *For any $\zeta \in \mathcal{Z}$, we have*

$$\begin{aligned} \mathsf{f}_{\mathcal{D},\zeta}(C) = &\ \min\{\mathrm{d}(C') \mid C' \in \mathcal{C}, \phi_\zeta(C') = \phi_\zeta(C)\}\,, \\ \mathsf{f}_{\mathcal{D},\zeta}(C) = &\ \mathsf{f}_{\mathcal{D}}(C) = \max_{\zeta' \in \mathcal{Z}} \mathsf{f}_{\mathcal{D},\zeta'}(C) = \end{aligned}$$

$$\max_{1 \leq i \leq r}\{\min\{\mathrm{d}(C') \mid C' \in \mathcal{C}, S_{C',\beta} = S_{C,\alpha_i}, \alpha_i = \zeta(\chi(C), n), \beta = \zeta(\chi(C'), n)\}\}\,,$$

where $C \in \mathcal{C}_{q,n}$ and $\alpha_1, \ldots, \alpha_r$ are all primitive n-th roots of unity over $\mathbb{F}_q$.

Unfortunately, the optimal root bound $\mathsf{f}_{\mathcal{D}}$ is not tight, as we claimed in the next theorem.

Theorem 3.1.20.

$$\mathsf{f}_{\mathcal{D}} \neq \mathrm{d}\,.$$

Proof. To prove our claim we need to find a code C where $\mathsf{f}_{\mathcal{D}}(C) \neq \mathrm{d}(C)$. Since it is not clear how to compute $\mathsf{f}_{\mathcal{D}}$, we divide the proof into two parts: one, where we suppose that we have two codes with some properties and we use them to prove our claim, and another, where we provide explicitly the above-mentioned codes.

Part I

We will provide in the second part of the proof two fields, $\mathbb{F}_{q_1}$ and $\mathbb{F}_{q_2}$, and a number $n \geq 1$, s.t.

- the two fields have different characteristics, which we may call p_1 for q_1 and p_2 for q_2,
- $\alpha_1, \ldots, \alpha_r$ are all the primitive n-th roots of unity over $\mathbb{F}_{q_1}$ and $\beta_1, \ldots, \beta_r$ are all the primitive n-th roots of unity over $\mathbb{F}_{q_2}$.

We take any $\zeta_1, \ldots, \zeta_r \in \mathcal{Z}$ s.t. $\zeta_i(p_1, n) = \alpha_i$ and $\zeta_i(p_2, n) = \beta_i$, for $1 \leq i \leq r$ (this is always possible). We will also provide two cyclic codes of length n, C_1 and C_2, the former over $\mathbb{F}_{q_1}$ and the latter over $\mathbb{F}_{q_2}$, s.t.

$$\mathrm{d}(C_1) < \mathrm{d}(C_2) \tag{3.7}$$

$$S_{C_1,\alpha_i} = S_{C_2,\beta_i}, \; 1 \leq i \leq r \, . \tag{3.8}$$

Observe that (3.8) implies $\phi_{\zeta_i}(C_1) = \phi_{\zeta_i}(C_2)$ for any i. We denote by S_i the set S_{C_1,α_i}, for any i. We have by Corollary 3.1.19 that $\mathsf{f}_{\mathcal{D}}(C_2)$ equals

$$\max_{1 \leq i \leq r} \{\min\{\mathrm{d}(C') \mid C' \in \mathcal{C}, S_{C',\beta} = S_{C_2,\alpha_i}, \alpha_i = \zeta_i(\ldots), \beta = \zeta_i(\ldots)\}\} \, , \tag{3.9}$$

which may be written as

$$\mathsf{f}_{\mathcal{D}}(C_2) = \max_{1 \leq i \leq r} \{\min T^{\zeta_i}_{(n,S_i)}\} \, . \tag{3.10}$$

By (3.8), for any i, we have $C_1 \in V^{\zeta_i}_{(n,S_i)}$ and hence $\mathrm{d}(C_1) \in T^{\zeta_i}_{(n,S_i)}$, which means

$$\min T^{\zeta_i}_{(n,S_i)} \; \leq \; \mathrm{d}(C_1) \, . \tag{3.11}$$

Putting together (3.11), (3.10) and (3.7), we get

$$\mathsf{f}_{\mathcal{D}}(C_2) = \max_{1 \leq i \leq r} \{\min T^{\zeta_i}_{(n,S_i)}\} \leq \max_{1 \leq i \leq r} \mathrm{d}(C_1) = \mathrm{d}(C_1) < \mathrm{d}(C_2) \, ,$$

which shows $\mathsf{f}_{\mathcal{D}}(C_2) < \mathrm{d}(C_2)$ and proves our claim.

Part II

It is enough to take $\mathbb{F}_{q_1} = \mathbb{F}_3$, $\mathbb{F}_{q_2} = \mathbb{F}_{17}$ and $n = 16$. There are $r = 8$ primitive n-th roots of unity. As cyclic codes C_1 and C_2, we take two codes with the same defining set $S = S_{C_1,\alpha_1} = S_{C_2,\beta_1}$,

$$S = \{1, 2, 3, 4, 6, 9, 11, 12\} \, ,$$

but note that $S = S_{C_1,\alpha_1}$ is the union of three cyclotomic sets over $\mathbb{F}_2$, while $S = S_{C_2,\beta_1}$ is the union of eight cyclotomic set over $\mathbb{F}_{17}$.
A quick computation (see Section 9.3) shows that $\mathrm{d}(C_1) = 5$ and $\mathrm{d}(C_2) = 6$.

□

What we call root bounds are sometimes called “BCH-like” bounds, since they include the BCH bound and its generalizations (the Hartmann-Tzeng bound, the Roos bound, etc.). In Subsection 3.4 and 3.5 we will see exactly what known bounds fall within our class.

We believe that the implications of Theorem 3.1.20 are noteworthy. Theorem 3.1.20 states that if you get a bound which depends only the information given by the defining sets, it does not matter how smart you are and how computationally costly is your bound, you will never get the distance for all cyclic codes. In other words, if you want actual improvements on known BCH-like bounds, you should try to use other information apart from defining sets.
There are two interesting questions naturally raised by Theorem 3.1.20. The first concerns the practical computation of f (and $\mathsf{f}_{\mathcal{D}}$). Apparently, computing f using either Definition 3.1.16 or (3.4) requires an unspecified number of computations. In principle, one should go through all $f \in \mathcal{R}$ (in the former case), which are infinite, or through all fields coprime with the length, which again are infinite. On the other hand, for any given code the value to be computed is finite and bounded by the distance, so it is obvious that the right value would be found after checking a finite number of $f \in \mathcal{R}$ (or of fields). However, we would not able to realize when we reach our value, unless infinite computations are performed. An `effective` algorithm is usually defined as an algorithm that runs in a finite and a priori bounded time (e.g., polynomial-time algorithms, exponential-time algorithms). Computing f from Definition 3.1.16 or (3.4) is non-effective (and useless in practice). The problem to compute f (or $\mathsf{f}_{\mathcal{D}}$) in a finite time will be faced in Chapter 8.

The second question comes from the proof of Theorem 3.1.20. The proof requires two codes with the same defining set and length, but over fields of different characteristic (otherwise they would have the same distance, due to Proposition 3.1.8). Thus the following question remains open.

Problem 3.1.21. Is there a finite field $\mathbb{F}_q$ s.t. $\mathsf{f}_{\mathcal{D}}$ is tight on

$$\mathcal{C}_q = \bigcup_{n\geq 1,(n,q)=1} \mathcal{C}_{q,n} \quad ?$$

3.2 Root bounds and $\mathcal{U}$

To determine the rank of a matrix in $\mathcal{U}$, as defined in Definition 2.2.10, is a very difficult problem, since in principle you have to run through an infinite number of matrices in an infinite number of fields. On the other hand, Theorem 2.2.16 depends on this rank notion and is of a paramount importance within our theory. Fortunately, we do not need to determine precisely the rank, in order to apply said theorem, but we only need to lower-bound the rank.

In this subsection we propose a simple but powerful method to verify the linear independence of a set of r rows in $\mathcal{U}^n$. This method is called “single procedure“ in [Sal01], but we prefer to call it the “singleton procedure“ as in [BS06].

Finally, we prove that using the singleton procedure we are able not only to lower-bound the rank but also to reach it exactly.
We start with a few definitions and lemmas.

Definition 3.2.1. *Let A be a matrix, either over a field $\mathbb{K}$ or over $\mathcal{U}$. We denote the j-th column of A by $A[j]$ and the (i,j)-th entry of A by $A[i,j]$.*

Let M be a matrix over $\mathcal{U}$. We say that $M[j]$ is a `singleton` *if it has only one non-zero component $M[i,j]$, i.e. $M[i,j] = \Delta^{+}$ and $M[l,j] = 0$ for $l \neq i$. When this happens, we say that the $i-th$ row is the row* `corresponding` *to the singleton.*

Singletons play a special role, thanks to the following two lemmas.

Lemma 3.2.2. *Let M be a matrix over $\mathcal{U}$ and $M[j]$ be one of its columns. If $M[j]$ is a singleton, then the corresponding row is linearly independent from the others.*

Proof. Let us suppose that $M \in \mathcal{U}^{m\times n}$. Let $r^{[1]}, \dots, r^{[m]}$ be the rows of M and let $r^{[i]}$ be the row corresponding to the singleton. If $r^{[i]}$ is a linear combination of $r^{[1]}, \dots, r^{[i-1]}, r^{[i+1]}, \dots, r^{[m]}$, there are a field $\mathbb{K}$, instantiations $\bar{r}^{[k]} = (\bar{r}^{[k]}_1, \dots, \bar{r}^{[k]}_n) \in \mathtt{In}(r^{[k]}, \mathbb{K})$, $1 \leq k \leq m$, and scalars not all zero $\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_m \in \mathbb{K}$ such that:

$$\lambda_1 \bar{r}^{[1]} + \cdots + \lambda_{i-1}\bar{r}^{[i-1]} + \lambda_{i+1}\bar{r}^{[i+1]} + \cdots + \lambda_m \bar{r}^{[m]} = \bar{r}^{[i]}.$$

In particular, we have: $\lambda_1 \bar{r}^{[1]}_j + \cdots + \lambda_{i-1}\bar{r}^{[i-1]}_j + \lambda_{i+1}\bar{r}^{[i+1]}_j + \cdots + \lambda_n \bar{r}^{[n]}_j = \bar{r}^{[i]}_j$. But by hypothesis $\bar{r}^{[k]}_j = 0$ for $k \neq i$ while $\bar{r}^{[k]}_j \neq 0$, so we have a contradiction. □

Lemma 3.2.3. *Let M be a matrix over $\mathcal{U}$ and $M[j]$ be one of its columns. Suppose $M[j]$ is a singleton and let row i be its corresponding row. Let M' be the matrix obtained from M by erasing column j and row i. Then M has full rank if and only if M' has full rank.*

Proof. We can suppose $M \in \mathcal{U}^{m\times n}$, so M has the form

$$M = \begin{pmatrix} a_{1,1} & \cdots & a_{1,(j-1)} & 0 & a_{1,(j+1)} & \cdots & a_{1,n} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{i,1} & \cdots & a_{i,(j-1)} & \Delta^{+} & a_{i,(j+1)} & \cdots & a_{i,n} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{m,1} & \cdots & a_{m,(j-1)} & 0 & a_{m,(j+1)} & \cdots & a_{m,n} \end{pmatrix}.$$

By s- deleting the singleton we obtain the matrix

$$M' = \begin{pmatrix} a_{1,1} & \cdots & a_{1,(j-1)} & a_{1,(j+1)} & \cdots & a_{1,n} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ a_{i-1,1} & \cdots & a_{i-1,(j-1)} & a_{i-1,(j+1)} & \cdots & a_{i-1,n} \\ a_{i+1,1} & \cdots & a_{i+1,(j-1)} & a_{i+1,(j+1)} & \cdots & a_{i+1,n} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ a_{m,1} & \cdots & a_{m,(j-1)} & a_{m,(j+1)} & \cdots & a_{m,n} \end{pmatrix}.$$

If M' does not have full rank then there exist $m-1$ scalars $\lambda_1, \dots, \lambda_{m-1}$ in a field $\mathbb{K}$ and vectors $v^{[1]}, \dots, v^{[m-1]} \in \mathbb{K}^n$, for any $1 \leq k \leq m-1$, where $v^{[k]} = (v_1^{[k]}, \dots, v_n^{[k]})$ is an instance of the k-th row of M' over $\mathbb{K}$, such that:

- $(\lambda_1, \dots, \lambda_{m-1}) \neq (0, \dots, 0)$
- $\sum_{k=1,}^{m-1} \lambda_k v^{[k]} = 0$.

We denote with $\bar{M}$ the matrix obtained removing the $i-$th row from M and we define $\bar{v}^{[1]}, \dots, \bar{v}^{[m-1]} \in \mathbb{K}^n$ as:

$$\bar{v}_t^{[k]} = \begin{cases} v_t^{[k]} & \text{if } t \neq j \\ 0 & \text{otherwise} \end{cases} \quad \text{, for any } 1 \leq k \leq m-1, \ 1 \leq t \leq n.$$

With this definition $\bar{v}^{[k]}$ is obviously an instance of the k-th row of $\bar{M}$, and $\sum_{k=1}^{m-1} \lambda_k \bar{v}^{[k]} = 0$, so that $\bar{M}$ has not full rank. But this means that also M has not full rank, which is a contradiction. □

Note that both for Lemma 3.2.2 and Lemma 3.2.3 also a trivial proof is achievable, by noting that their statement is true if "translated" over any field.

We are ready to describe our `singleton procedure`.

We start from a set of r rows of length n, with $r \leq n$, and we want to test whether they are linearly independent. We take our r rows to form a matrix $A_r \in \mathcal{U}^{r \times n}$. We search for a singleton in A_r. If column $A_r[j]$ is a singleton, we know that the corresponding row is linearly independent from the others (Lemma 3.2.2). Then we erase from A_r the $j-th$ column and the corresponding row (we call this operation `s-deletion`). We denote by A_{r-1} the $(r-1) \times (n-1)$ matrix so obtained. Matrix A_{r-1} has full rank if and only if A_r has (Lemma 3.2.3).

We search for a new singleton in A_{r-1} and proceed as before. If this procedure can continue until we have obtained a $1 \times (n-r+1)$ matrix A_1 containing at least one Δ^+, then the initial matrix A_r has full rank, since A_1 has. In this case we say that the singleton procedure is `successful` for the original set of r rows. However, if we cannot find a singleton either in A_r or in any successive A_i, then we say that the singleton procedure is `not successful`.

We provide an example.

Example 3.2.4.

$$A_3 = \begin{pmatrix} 0 & \Delta^{+} & \Delta & 0 \\ 0 & 0 & \Delta^{+} & \Delta^{+} \\ \Delta^{+} & 0 & 0 & \Delta \end{pmatrix} \to A_2 = \begin{pmatrix} 0 & \Delta^{+} & \Delta^{+} \\ \Delta^{+} & 0 & \Delta \end{pmatrix}, \begin{array}{c} j=2 \\ i=1 \end{array}$$

$$A_2 = \begin{pmatrix} 0 & \Delta^{+} & \Delta^{+} \\ \Delta^{+} & 0 & \Delta \end{pmatrix} \to A_1 = (\Delta^{+}, \Delta^{+}), \begin{array}{c} j=1 \\ i=2 \end{array},$$

hence the singleton procedure is successful for A_3.

Remark 3.2.5. Let $M \in \mathcal{U}^{m\times n}$, without loss of generality $m \leq n$. When we apply the singleton procedure, for each s-deletion we erase one column and one row. So to say that the singleton procedure is successful for M is equivalent to finding a square $m \times m$ submatrix of M for which the singleton procedure is successful.

We can summarize our arguments in the next proposition.

Theorem 3.2.6. *If the singleton procedure is successful for a set of rows, then they are linearly independent over $\mathcal{U}$.*

Proof. This follows from Lemma 3.2.2, Lemma 3.2.3 and the obvious fact that the last matrix A_1, is linearly independent. □

Theorem 3.2.6 will be our preferred tool to give formal proofs for bounds since it allows us to give estimates on the rank of a matrix over $\mathcal{U}$.

Definition 3.2.7. *Given a matrix M be a matrix over $\mathcal{U}$, we denote by* $\mathrm{prk}(M)$ *the **pseudo-rank** of M, i.e. the largest t such that there exists a set of t rows in M for which the singleton procedure is successful.*

Remark 3.2.8. For the moment, we can only say that $\mathrm{prk}(M) \leq \mathrm{rk}(M)$, because if the singleton procedure is not successful for a set of rows, then we cannot conclude they are linearly dependent over $\mathcal{U}$. However we will show in Theorem 3.2.18 that rank and pseudo-rank coincide.

In some simple cases we can establish equalities between ranks and pseudo-ranks of different matrices. This is done in the following lemmas. Observe that Lemma 3.2.9, Lemma 3.2.10 and 3.2.12 are obvious, since their "translation" over any field holds.

Lemma 3.2.9. *Let $\mathbf{u}, \mathbf{v} \in \mathcal{U}^n$. Let $m \in \mathbb{N}$. If $\mathbf{u}$ is obtained from a shift of $\mathbf{v}$ by m places, then*

$$\mathrm{rk}(M(\mathbf{u})) = \mathrm{rk}(M(\mathbf{v})) \qquad\qquad \mathrm{prk}(M(\mathbf{u})) = \mathrm{prk}(M(\mathbf{v})).$$

Lemma 3.2.10. *Let $\sigma \in \mathrm{Sym}(n)$ a permutation. Let $M \in \mathcal{U}^{n\times n}$ and M' be the matrix obtained by applying σ to the rows (resp. columns) of M. Then*

$$\mathrm{rk}(M) = \mathrm{rk}(M') \qquad\qquad \mathrm{prk}(M) = \mathrm{prk}(M').$$

Definition 3.2.11. *Let $\mathbf{u} \in \mathcal{U}^n$. We denote by $\hat{\mathbf{u}}$ the* ***reflection*** *of $\mathbf{u}$, i.e. the vector in $\mathcal{U}$ s.t. $\quad \mathbf{u}[i] = \hat{\mathbf{u}}[n-i+1]$ for any $1 \le i \le n$.*
Similarly, we denote by $\hat{M}$ the reflection of $M \in \mathcal{U}^{n\times n}$, i.e. the matrix such that

$$M[j,i] = \hat{M}[j, n-i+1], \qquad 1 \le i \le n, \quad 1 \le j \le n\,.$$

Lemma 3.2.12. *For any $M \in \mathcal{U}^{n\times n}$, we have*

$$\mathrm{rk}(M) = \mathrm{rk}(\hat{M}) \qquad\qquad \mathrm{prk}(M) = \mathrm{prk}(\hat{M}).$$

Lemma 3.2.13. *For any $\mathbf{u} \in \mathcal{U}^n$, we have*

$$\mathrm{rk}(M(\mathbf{u})) = \mathrm{rk}(M(\hat{\mathbf{u}})) \qquad\qquad \mathrm{prk}(M(\mathbf{u})) = \mathrm{prk}(M(\hat{\mathbf{u}})).$$

Proof. For any $1 \le i,j \le n$, $i \ne j$, let $(i\ j)$ be a transposition in the symmetric group $\mathrm{Sym}(n)$. Consider the permutation:

$$\sigma = \prod_{i=2}^{\lfloor \frac{n}{2} \rfloor} (i \quad n-i+2) \ \in \ \mathrm{Sym}(n).$$

The matrix $M(\hat{\mathbf{u}})$ is obtained by applying σ to the rows of $\hat{M}(\mathbf{u})$ and so we may apply Lemma 3.2.10 and Lemma 3.2.12. □

The following proposition establishes an important rank bound, which will be often used in proofs.

Proposition 3.2.14. *Let $A = M(\mathbf{v}) \in \mathcal{U}^n$ be a circulant matrix, and let $r \ge 0$ be an integer. If $\mathbf{v}$ has the form*

$$\mathbf{v} = (\overbrace{0, \ldots, 0}^{r}, \Delta^{+}, *, \ldots, *)\,,$$

where $*$ denotes any element of $\mathcal{U}$, then $\mathrm{rk}(A) \ge \mathrm{prk}(A) \ge r+1$.

Proof. Let $A_{r+1} \in \mathcal{U}^{(r+1)\times n}$ be the matrix obtained by the first $r+1$ rows of $M(\mathbf{v})$. By induction on r we show that the singleton procedure is successful for A_{r+1}. If $r = 0$, it is clear that the singleton procedure is successful since A_{r+1} coincide with $\mathbf{v}$ and $\mathbf{v}[r+1] = \Delta^{+}$.

Let $r > 0$. Matrix A_{r+1} has the form:

$$A_{r+1} = \begin{pmatrix} 0 & \dots & 0 & 0 & \Delta^+ & \dots & \\ \Delta & 0 & \dots & 0 & 0 & \Delta^+ & \dots \\ \vdots & & \ddots & & \vdots & & \\ \vdots & & & \ddots & & & \\ \Delta & \dots & & \Delta & 0 & 0 & \dots \end{pmatrix}$$

where, with abuse of notation, we have put a Δ in all entries for which we have no information on the value. Column $A_{r+1}(\mathbf{v})[r+1]$ is clearly a singleton, since $A(\mathbf{v})[1, r+1] = \mathbf{v}[r+1] = \Delta^+$, and, for any $2 \leq i \leq r+1$, we have:

$$A_{r+1}[i, r+1] = A_{r+1}[i-1, r] = \dots$$

$$\dots = A_{r+1}[1, r-i+2] = \mathbf{v}[r-i+2] = 0.$$

Then we can erase the first row and the $(r+1)-$th column to obtain a matrix A_r, that corresponds exactly to the first r rows of a matrix $M(\mathbf{v}')$, with $\mathbf{v}'$ of the form:

$$(\overbrace{0, \dots, 0}^{r-1}, \Delta^+, *, \dots, *).$$

By induction hypothesis the singleton procedure is successful for A_r, which implies that it is successful for A_{r+1}, too. □

The following proposition is a key step in proving that the pseudo-rank and the rank coincide for any matrix over $\mathcal{U}$.

Proposition 3.2.15. *Let $A_n = \{ a_{ij} \}_{1 \leq i,j \leq n} \in \mathcal{U}^{n \times n}$ be a $n \times n$ square matrix on $\mathcal{U}$. If $\operatorname{rk}(A_n) = n$ then A_n has a singleton.*

Proof. Note that A_n has to contain at least a Δ^+ for each column and each row, otherwise there is an instance of one of its columns that is zero, so that the rank will not be n. In particular this proves the proposition when $n = 1$. Let us suppose $n \geq 2$. By contradiction we suppose that A_n has no singletons.
Let $C_1 = \{ 1 \leq j \leq n \mid a_{1,j} \neq 0 \}$ be the set of entries in the first row of A_n which are Δ or Δ^+. We define for any $j \in C_1$ the set $R^1_j = \{ 1 \leq k \leq n \mid a_{k,j} \neq 0 \}$ of entries in the $j-$th column which are Δ or Δ^+. By hypothesis, for any $j \in C_1$, we have $|R^1_j| \geq 2$ and in particular there exists $\bar{k} \in R^1_j$ such that $a_{\bar{k},j} = \Delta^+$. We define inductively for any $2 \leq i \leq n$ the sets

$$C_i = \{ 1 \leq j \leq n \mid a_{i,j} \neq 0 \} \setminus \cup_{k=1}^{i-1} C_k$$

and for any $j \in C_i$,
$$R^i_j = \{\, 1 \leq k \leq n \mid a_{k,j} \neq 0 \,\} \,.$$
Note that $\sqcup_{i=1}^n C_i = \{\, 1, \ldots, n \,\}$ since any column contains at least a Δ^+ by hypothesis, and $C_i \cap C_j = \emptyset$ if $i \neq j$, because they are disjoint by construction. Observe that if $C_i = \emptyset$ then for any j we have $R^i_j = \emptyset$ while. If $C_i \neq \emptyset$ then $i \in R^i_j$ for any $j \in C_i$, by definition and $|R^i_j| \geq 2$, since we assume that A_n has no singletons.

Let p be a prime, $p > n$. We provide n vectors, $v^{[1]}, \ldots, v^{[n]}$ in $(\mathbb{F}_p)^n$, which are instantiations of the rows of A_n such that they are linear dependent, so that A_n cannot have rank n. For any $\overline{j} \in \{\, 1, \ldots, n \,\}$ let C_r be such that $\overline{j} \in C_r$. We define for $1 \leq k \leq n$:
$$v^{[k]}_{\overline{j}} = \begin{cases} 0, & \text{if } k \notin R^r_{\overline{j}}, \\ 1, & \text{if } k \in R^r_{\overline{j}} \text{ and } k \neq r, \\ p - (|R^r_{\overline{j}}| - 1), & \text{if } k = r. \end{cases} \tag{3.12}$$
We have to prove that:

(a) $v^{[k]}$ is an instantiation of the $k-th$ row of A_n

(b) there exist $\lambda_1, \ldots, \lambda_n \in \mathbb{F}_p$ such that $(\lambda_1, \ldots, \lambda_n) \neq (0, \ldots, 0)$ and $\sum_{k=1}^n \lambda_k v^{[k]} = 0$.

We start with (a). Let k be any element of $\{\, 1, \ldots, n \,\}$, we have to prove that for $1 \leq j \leq n$:

- if $a_{k,j} = 0$ then $v^{[k]}_j = 0$,
- if $a_{k,j} = \Delta^+$ then $v^{[k]}_j \neq 0$.

For any j, let r_j be such that $j \in C_{r_j}$. If $a_{k,j} = 0$ then $k \notin R^{r_j}_k$ and thanks to (3.12) we have $v^{[k]}_j = 0$. Similarly, if $a_{k,j} = \Delta^+$, then $k \in R^{r_j}_k$ and we have to consider two cases: $k = r_j$ or $k \neq r_j$. If $k \neq r_j$ then $v^{[k]}_j = 1$, else if $k = r_j$ then $v^{[k]}_i = p - (|R^r_{\overline{j}}| - 1)$ and since $1 \leq p - n + 1 \leq p - (|R^r_{\overline{j}}| - 1) \leq p - 1$, thus $p - (|R^r_{\overline{j}}| - 1) \neq 0$ and so we have $v^{[k]}_j \neq 0$. Hence (a) is proved.

To prove (b) we claim that for any $1 \leq j \leq n$: $\sum_{k=1}^n v^{[k]}_j = 0$. Let us fix any j, and let r be such that $j \in C_r$. Then
$$\sum_{k=1}^n v^{[k]}_j = \sum_{k \in R^r_j} v^{[k]}_j = v^{[r]}_j + \sum_{k \in R^r_j, k \neq r} v^{[k]}_j = p - (|R^r_j| - 1) + (|R^r_j| - 1) = 0.$$

□

The proof of Proposition 3.2.15 is rather technical and not easy to follow. So we provide here an example to clarify its details.

Example 3.2.16. Let us consider a 4×4 matrix over $\mathcal{U}$, A_4, which has no singleton and we provide instances of the rows which are linearly dependent over $\mathbb{Z}_5$.

$$A_4 = \begin{pmatrix} 0 & \Delta^{+} & \Delta & 0 \\ \Delta^{+} & \Delta & 0 & 0 \\ 0 & 0 & \Delta & \Delta^{+} \\ \Delta^{+} & 0 & \Delta^{+} & \Delta^{+} \end{pmatrix}$$

The sets that we consider are: $C_1 = \{ 2, 3 \}$, $C_2 = \{ 1 \}$, $C_3 = \{ 4 \}$, $C_4 = \emptyset$ and $R_2^1 = \{ 1, 2 \}$, $R_1^2 = \{ 2, 4 \}$, $R_4^3 = \{ 3, 4 \}$, $R_3^1 = \{ 1, 3, 4 \}$. We show how to choose $v^{[1]}$ $(k = 1)$. We have:

- $1 \in C_r = C_2$ and $1 \notin R_1^2 \implies v_1^{[1]} = 0$
- $2 \in C_r = C_1$, $1 \in R_2^1$ and $r = k \implies v_2^{[1]} = 5 - (|R_2^1| - 1) = 4$
- $3 \in C_r = C_1$, $1 \in R_3^1$ and $r = k \implies v_3^{[1]} = 5 - (|R_3^1| - 1) = 3$
- $4 \in C_r = C_3$ and $1 \notin R_4^3 \implies v_4^{[1]} = 0.$

Hence $v^{[1]} = (0, 4, 3, 0)$. Similarly, for $v^{[2]}$ $(k = 2)$:

- $1 \in C_r = C_2$, $2 \in R_1^2$ and $r = k \implies v_1^{[1]} = 5 - (|R_1^2| - 1) = 4$
- $2 \in C_r = C_1$, $2 \in R_2^1$ and $r \neq k \implies v_2^{[1]} = 1$
- $3 \in C_r = C_1$, $2 \notin R_3^1 \implies v_3^{[1]} = 0$
- $4 \in C_r = C_3$ and $2 \notin R_4^3 \implies v_4^{[1]} = 0.$

which gives $v^{[2]} = (4, 1, 0, 0)$. Doing the same also for $v^{[3]}$ and $v^{[4]}$ we obtain:

$$v^{[1]} = (0, 4, 3, 0) \qquad v^{[2]} = (4, 1, 0, 0) \qquad v^{[3]} = (0, 0, 1, 4) \qquad v^{[4]} = (1, 0, 1, 1).$$

Finally, note that they are instances of the rows of A_4 over $\mathbb{Z}_5$ and

$$\begin{pmatrix} 0 \\ 4 \\ 3 \\ 0 \end{pmatrix} + \begin{pmatrix} 4 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Proposition 3.2.17. *Let $A_n = \{ a_{i,j} \}_{1 \le i,j \le n}$ be in $\mathcal{U}^{n \times n}$. The following are equivalent:*

1. $\mathrm{rk}(A_n) = n$
2. *the singleton procedure is successful for A_n*

Proof.

(2) $\implies$ (1): see Theorem 3.2.6.

(1) $\implies$ (2): by induction. If $n = 1$ then thanks to Proposition 3.2.15 A_1 has a singleton, which means $A_1 = \left(\Delta^{+}\right)$ and the singleton procedure is trivially successful. Let us suppose $n > 1$. Thanks to Proposition 3.2.15 A_n has a singleton. The submatrix A_{n-1} obtained by s-deletion of the singleton of A_n has full rank, thanks to Lemma 3.2.3. By the iterative definition of singleton procedure, if the singleton procedure is successful for A_{n-1} then the singleton procedure is successful for A_n. By induction hypothesis, the singleton procedure is successful for A_{n-1} and so our claim follows.

□

Theorem 3.2.18. *Let M be any matrix over $\mathcal{U}$, then* $\mathrm{rk}(M) = \mathrm{prk}(M)$.

Proof. We suppose $M \in \mathcal{U}^{m\times n}$ and $t = \mathrm{rk}(M)$. We have that $\mathrm{prk}(M) \le t$ and M contains a square submatrix $M_t \in \mathcal{U}^{t\times t}$ such that $\mathrm{rk}(M_t) = t$. Thanks to Proposition 3.2.17 the singleton procedure is successful for M_t and by Remark 3.2.5 the singleton procedure is successful on the t rows of M corresponding to M_t. Thus, $\mathrm{prk}(M) \ge \mathrm{prk}(M_t) = t$.

□

3.3 Strict root bounds

Using bounds on ranks over $\mathcal{U}$, we are able to prove bounds on the distance, as we will see in this section. However, we will show that any bound of this type is actually a root bound, but not any root bound is of this type. To provide precise statements, we need a few definitions and results. Here we depart form the notation in [BS07] since we prefer to write “strict root” rather than “strong root”.

Definition 3.3.1. *A* `strict root function` *is a map $f : \mathcal{D} \to \mathbb{N}$ such that:*

$$\forall (n, S) \in \mathcal{D}, \quad f(n, S) \le \min\{\mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n, S))\} \tag{3.13}$$

We denote by $\mathcal{R}^S$ the class of all strict root functions.

We can remove any ambiguity from the term “strict root function”.

Proposition 3.3.2. *Any strict root function is a root function, that is $\mathcal{R}^S \subseteq \mathcal{R}$.*

Proof. Let f be a strict root function. We have to verify (3.2). Let $\zeta \in \mathcal{Z}$ and $C \in \mathcal{C}$. Let $p = \chi(C)$ and $\alpha = \zeta(p, n)$. We have

$$f \circ \phi_\zeta(C) = f(n, S_{C,\alpha})$$

Since f is a strict root function, we have (3.13), i.e.

$$f(n, S_{C,\alpha}) \leq \min\{\mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n, S_{C,\alpha}))\},$$

but the right-hand side is not bigger than $d(C)$, by Theorem 2.2.16. Putting all together, we get

$$f \circ \phi_\zeta(C) \leq d(C).$$

□

We propose some previous constructions introduced for root functions, which can be specialized in the case of strict root functions.

Proposition 3.3.3. *Let f be any strict root function. Then:*

1. *$f^\# \in \mathcal{R}^S$,*
2. *$f^\#$ is invariant,*
3. *$f \leq f^\#$,*
4. *$f^\#_{\mathcal{D}} = \max_{\zeta \in \mathcal{Z}} f_{\mathcal{D},\zeta}$.*

Proof. We only provide the proof of **1)**. The proofs of **2)**, **3)** and **4)**, are an easy adaption of the proof of Proposition 3.1.5.

Let $(n, S) \in \mathcal{D}$, what we have to prove is that $f^\#(n,S) \leq \min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n,S))\,\}$. We claim that for any $T \in (n,S)^\#$ it holds:

$$\min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n,S))\,\} = \min\{\, \mathrm{rk}(M(\mathbf{v})) \mid \mathbf{v} \in \mathcal{A}(R(n,T))\,\}.$$

Let $\zeta \in \mathcal{Z}$. Let $C, D \in \mathcal{C}_{q,n}$ be two naturally equivalent codes such that $\phi_\zeta(C) = (n,S)$ and $\phi_\zeta(D) = (n,T)$. Let $q \in \mathbb{N}$ such that $C, D \in \mathcal{C}_{q,n}$. Let g_C, $g_D \in \mathbb{F}_q[x]$ be the generator polynomials of C and D, respectively. Then, from Proposition 2.1.7, there is a permutation matrix P_λ such that $M(\mathrm{DFT}(g_C)) = P_\lambda M(\mathrm{DFT}(g_D))P_\lambda^T$. We observe that, from Definition 2.2.5, $R(n,S)$ is the vector $(u_0, \dots, u_{n-1})$ such that $u_i = 0$ if $g_C(\alpha^i) = 0$, where $\alpha = \zeta(n, \chi(C))$, and $u_i = \Delta$ otherwise. Similarly, $R(n,T)$ is the vector $(v_0, \dots, v_{n-1})$ such that $v_i = 0$ if $g_D(\alpha^i) = 0$ and $v_i = \Delta$ otherwise. Thus $M(R(n,S)) = P_\lambda M(R(n,T))P_\lambda^T$ and for any $\mathbf{u} \in \mathcal{A}(R(n,S))$ there is $\mathbf{v} \in \mathcal{A}(R(n,T))$ such that $M(\mathbf{u}) = P_\lambda M(\mathbf{v})P_\lambda^T$. Since P_λ and P_λ^T are permutations of rows or columns, thanks to Lemma 3.2.10, we conclude that $\mathrm{rk}(M(\mathbf{u})) = \mathrm{rk}(P_\lambda M(\mathbf{v})P_\lambda^T) = \mathrm{rk}(M(\mathbf{v}))$. So

$$\begin{aligned}\min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n,S))\,\} &= \min\{\mathrm{rk}(P_\lambda M(\mathbf{v})P_\lambda^T) \mid \mathbf{v} \in \mathcal{A}(R(n,T))\} \\ &= \min\{\mathrm{rk}(M(\mathbf{v})) \mid \mathbf{v} \in \mathcal{A}(R(n,T))\}.\end{aligned}$$

We are now able to prove **1)**. By definition of $f^{\#}$ we have $f^{\#}(n,S) = f(n,T)$ for at least one $T \in (n,S)^{\#}$, hence:

$$\begin{aligned} f^{\#}(n,S) = f(n,T) &\leq \min\{\, \mathrm{rk}(M(\mathbf{v})) \mid \mathbf{v} \in \mathcal{A}(R(n,T)) \,\} \\ &= \min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n,S)) \,\}\,. \end{aligned}$$

□

Proposition 3.3.4. *Let f be any strict root function. Then:*

1. *f^* is a strict root function,*
2. *f^* is monotone,*
3. *$f \leq f^*$,*
4. *if g is any monotone strict root function s.t. $f \leq g$, then $f^* \leq g$.*

Proof. We only provide the proof of **1)**. The proofs of **2)**, **3)** and **4)** are similar to those of Proposition 3.1.11. Let $(n,S) \in \mathcal{D}$. What we have to prove is that $f^*(n,S) \leq \min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n,S)) \,\}$.

By definition of f^*, we have that $f^*(n,S) = f(n,S')$ for some $S' \subseteq S$. If $S' \subseteq S$, we have $\mathcal{A}(R(n,S')) \supseteq \mathcal{A}(R(n,S))$, so:

$$\begin{aligned} f^*(n,S) = f(n,S') &\leq \min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n,S')) \,\} \\ &\leq \min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n,S)) \,\}\,. \end{aligned}$$

□

In the context of strict root bounds we can introduce another notion of maximality.

Definition 3.3.5. *If δ is a root bound associated to a strict root function, we say that δ is a strict root bound.*

We denote by $\mathcal{R}_D^S$ the class of all strict root bounds.

Clearly $\mathcal{R}_{\mathcal{D}}^S \subseteq \mathcal{R}_{\mathcal{D}}$.

We define a map f^S from $\mathcal{D}$ to $\mathbb{N}$ as follows

$$\mathsf{f}^S(n,S) = \max\{f(n,S) \mid f \in \mathcal{R}^S\}\,. \tag{3.14}$$

Theorem 3.3.6. *Map f^S is a strict root function, which is maximal in $\mathcal{R}^S$, monotone and invariant.*

Proof. We only prove that f^S is a strict root function. For the other claims it is enough to adapt the argument from the proof of Theorem 3.1.12, using Proposition 3.3.3 and Proposition 3.3.4.

Let (n, S) be any element of $\mathcal{D}$. By definition of f^S we have $\mathsf{f}^S(n, S) = f(n, S)$ for some $f \in \mathcal{R}^S$. Thus $\mathsf{f}^S(n, S) = f(n, S) \leq \min\{\, \mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n, S)) \,\}$.

□

From the definition of strict root functions, we get a characterization for the maximal strict root function.

Theorem 3.3.7.

$$\mathsf{f}^S(n, S) = \min\{\mathrm{rk}(M(\mathbf{u})) \mid \mathbf{u} \in \mathcal{A}(R(n, S))\}\,.$$

Moreover, for any $f \in \mathcal{R}$ we have $f \in \mathcal{R}^S$ if and only if $f \leq \mathsf{f}^S$.

Note that Theorem 3.3.7 and Theorem 3.2.18 obviously imply that bounded finite-time computations are enough to compute f^S.

Since f is maximal in $\mathcal{R}$, $\mathsf{f}^S \leq \mathsf{f}$. Actually we will see in Theorem 3.5.8 that $\mathsf{f}^S < \mathsf{f}$.

Let $i \geq 1$. We define three patterns of symbols which correspond to vectors in $\mathcal{U}^i$, sometimes called "blocks":

$$(0)^i = (\overbrace{0, \dots, 0}^{i}), \qquad (\Delta)^i = (\overbrace{\Delta, \dots, \Delta}^{i}), \qquad (\Delta^{\!\!+})^i = (\overbrace{\Delta^{\!\!+}, \dots, \Delta^{\!\!+}}^{i}),$$

$$(0)^0 = (\Delta)^0 = (\Delta^{\!\!+})^0 = \emptyset.$$

Using these three first blocks we can define multiple blocks using concatenation, for example $(0)^3(\Delta)^2 = (0, 0, 0, \Delta, \Delta)$ or $(0)^2(\Delta^{\!\!+})^4 = (0, 0, \Delta^{\!\!+}, \Delta^{\!\!+}, \Delta^{\!\!+}, \Delta^{\!\!+})$. We also define blocks of blocks, with an obvious meaning, as for example:

$$((0)^2(\Delta^{\!\!+})^3)^2(\Delta)^2 = (0, 0, \Delta^{\!\!+}, \Delta^{\!\!+}, \Delta^{\!\!+}, 0, 0, \Delta^{\!\!+}, \Delta^{\!\!+}, \Delta^{\!\!+}, \Delta, \Delta).$$

Let us consider two vectors of different length, for example:

$$\mathbf{u} = (\Delta^{\!\!+}, 0, \Delta) \in \mathcal{U}^3, \qquad \mathbf{v} = (\Delta^{\!\!+}, 0, \Delta, \Delta^{\!\!+}, \Delta^{\!\!+}, 0) \in \mathcal{U}^6.$$

Let $\mathbb{K}$ be any field, then the vector $\mathbf{u}$ represents a vector in $\mathbb{K}^3$ with the first coordinate different from zero, the second coordinate equal to zero and the third component that is any element of $\mathbb{K}$. In the same way, $\mathbf{v}$ represents a vector of $\mathbb{K}^6$ such that the first, the fourth and the fifth component are different from zero, the second component is zero and the third component is any element of $\mathbb{K}$.

We note that the constraints for the components of $\mathbf{u}$ coincide with the constraints for the first three components of $\mathbf{v}$ and in this case we write $\mathbf{u} \preccurlyeq \mathbf{v}$. The previous example shows a particular case of a special kind of relation among vectors over $\mathcal{U}$, that we are going to define in the following definition.

Definition 3.3.8. *Let $n, m \in \mathbb{N}$ such that $n \geq m$. Let π be the projection of $\mathcal{U}^n$ on $\mathcal{U}^m$ as follows:*

$$\pi \colon \mathcal{U}^n \to \mathcal{U}^m, \qquad \pi((v_1, \ldots, v_n)) = (v_1, \ldots, v_m).$$

Let $\mathbf{u} \in \mathcal{U}^m$ and $\mathbf{v} \in \mathcal{U}^n$, we write $\mathbf{u} \preccurlyeq \mathbf{v}$ if there is $0 \leq i \leq n-1$ such that

$$\mathcal{A}(\pi(\mathrm{sh}^i(\mathbf{v})) \subseteq \mathcal{A}(\mathbf{u}).$$

When $\mathbf{u} \preccurlyeq \mathbf{v}$ we say that $\mathbf{u}$ is ***`included`*** *in $\mathbf{v}$.*

Our Definition 3.3.8 of inclusion of vectors has some particular properties that we are going to show.

Proposition 3.3.9. *Let $\mathbf{u} \in \mathcal{U}^m$, $\mathbf{v} \in \mathcal{U}^n$, $\mathbf{w} \in \mathcal{U}^t$ with $m, n, t \geq 1$. We indicate with $\mathbf{uv}$ the vector in $\mathcal{U}^{m+n}$ obtained by concatenating $\mathbf{u}$ and $\mathbf{v}$, i.e. $\mathbf{uv} = (u_1, \ldots, u_m, v_1, \ldots, v_n)$. The following statements hold:*

a) $(\Delta) \preccurlyeq (\Delta\!\!^{+})$, $(\Delta\!\!^{+}) \preccurlyeq (\Delta)$, $(\Delta) \preccurlyeq (0)$, $(\Delta) \not\preccurlyeq (0)$.

b) $\mathbf{v} \preccurlyeq \mathbf{v}$.

c) $\mathbf{u} \preccurlyeq \mathbf{v} \iff \mathbf{u} \preccurlyeq \mathrm{sh}(\mathbf{v})$.

d) $\mathbf{v} \preccurlyeq \mathbf{uv}$, $\mathbf{v} \preccurlyeq \mathbf{vu}$.

e) $\mathbf{v} \preccurlyeq \mathbf{uvw}$.

f) $(\Delta)^m \preccurlyeq \mathbf{v}$ *for any* $\mathbf{v} \in \mathcal{U}^n$ *s.t.* $m \leq n$.

Proof.

a) Since $(\Delta), (\Delta\!\!^{+}), (0) \in \mathcal{U}^1$ the shift is trivial and then we can ignore it. We have: $\mathcal{A}((\Delta)) = \{\, (\Delta\!\!^{+}) \,\}$, $\mathcal{A}((\Delta\!\!^{+})) = \{\, (\Delta\!\!^{+}) \,\}$, $\mathcal{A}((0)) = \emptyset$,

$$\begin{aligned} \mathcal{A}((\Delta\!\!^{+})) &\subseteq \mathcal{A}((\Delta)), & \mathcal{A}((\Delta)) &\subseteq \mathcal{A}((\Delta\!\!^{+})), \\ \mathcal{A}((0)) &\subseteq \mathcal{A}((\Delta)), & \mathcal{A}((\Delta)) &\not\subseteq \mathcal{A}((0)). \end{aligned}$$

b) Since $n = m$ the projection becomes trivially the identity and it is sufficient to take $i = 0$ in order to have $\mathcal{A}(\pi(\mathrm{sh}^0(\mathbf{v}))) = \mathcal{A}((\mathbf{v})) \subseteq \mathcal{A}(\mathbf{v})$.

c) " $\Longrightarrow$ ". Let $\overline{\mathbf{v}} = \mathrm{sh}(\mathbf{v})$ and let $0 \leq i \leq n-1$ be s.t. $\mathcal{A}(\pi(\mathrm{sh}^i(\mathbf{v}))) \subseteq \mathcal{A}(\mathbf{u})$. Denoting $\overline{i} = (i-1)_n$ we have $\mathrm{sh}^{\overline{i}}(\overline{\mathbf{v}}) = \mathrm{sh}^i(\mathbf{v})$ and so $\mathcal{A}(\pi(\mathrm{sh}^{\overline{i}}(\overline{\mathbf{v}}))) \subseteq \mathcal{A}(\mathbf{u})$ which implies $\mathbf{u} \preccurlyeq \overline{\mathbf{v}}$. The proof of " $\Longleftarrow$ " is analogous.

d) Since $\mathrm{sh}^n(\mathbf{uv}) = \mathbf{vu}$, we have $\pi(\mathbf{vu}) = \mathbf{v}$, $\mathcal{A}(\pi(\mathrm{sh}^n(\mathbf{uv}))) = \mathcal{A}(\pi(\mathbf{vu})) = \mathcal{A}(\mathbf{v}) \subseteq \mathcal{A}(\mathbf{v})$. In the same way $\pi(\mathbf{vu}) = \mathbf{v}$ and $\mathcal{A}(\pi(\mathbf{vu})) = \mathcal{A}(\mathbf{v}) \subseteq \mathcal{A}(\mathbf{v})$.

e) From (d) we have that $\mathbf{v} \preccurlyeq \mathbf{vwu}$ for all $\mathbf{wu} \in \mathcal{U}^{t+m}$ and since $\mathbf{uvw} = \mathrm{sh}^m(\mathbf{vwu})$ we use (c) to conclude that $\mathbf{v} \preccurlyeq \mathbf{uvw}$.

f) We have $\mathcal{A}((\Delta)^m) = \left\{\, 0, \Delta^{+} \,\right\}^m \backslash \mathbf{0}$ and by Definition 2.2.15 $\mathcal{A}(\pi(\mathbf{v})) \subseteq \left\{\, 0, \Delta^{+} \,\right\}^m \backslash \mathbf{0}$ for any $\mathbf{v} \in \mathcal{U}^n$, $m \leq n$.

□

Example 3.3.10.

- $(\Delta, \Delta^{+}\Delta) \preccurlyeq (0, 0, \Delta, \Delta^{+}, \Delta, \Delta^{+})$ by Proposition 3.3.9 - (e), since $(\Delta, \Delta^{+}\Delta) \preccurlyeq (0, 0)(\Delta, \Delta^{+}, \Delta)(\Delta^{+})$;

- $(0)^2(\Delta) \preccurlyeq (0, \Delta, \Delta^{+}, \Delta, 0)$ by Proposition 3.3.9 - (c)-(d), since $(0)^2(\Delta) \preccurlyeq (0, 0)(\Delta, \Delta^{+}, \Delta)$, and we can obtain $(0, \Delta, \Delta^{+}, \Delta, 0)$ if we shift by $n - 1$ positions;

- $(0, \Delta^{+}, \Delta^{+}) \not\preccurlyeq (\Delta^{+}, \Delta^{+}, 0, \Delta, \Delta)$, because we have $\mathcal{A}\left((0, \Delta^{+}, \Delta^{+})\right) = \left\{\, (0, \Delta^{+}, \Delta^{+}) \,\right\}$, and for $0 \leq i \leq 4$:

$$i = 0 \quad \mathcal{A}(\pi(\Delta^{+}, \Delta^{+}, 0, \Delta, \Delta)) = \left\{\, (\Delta^{+}, \Delta^{+}, 0) \,\right\}$$

$$i = 1 \quad \mathcal{A}(\pi(\Delta, \Delta^{+}, \Delta^{+}, 0, \Delta)) = \left\{\, (\Delta^{+}, \Delta^{+}, \Delta^{+}), (0, \Delta^{+}, \Delta^{+}) \,\right\}$$

$$i = 2 \quad \mathcal{A}(\pi(\Delta, \Delta, \Delta^{+}, \Delta^{+}, 0)) = \left\{\, (\Delta^{+}, \Delta^{+}, \Delta^{+}), (0, 0, \Delta^{+}), (0, \Delta^{+}, \Delta^{+}), (\Delta^{+}, 0, \Delta^{+}) \,\right\}$$

$$i = 3 \quad \mathcal{A}(\pi(0, \Delta, \Delta, \Delta^{+}, \Delta^{+})) = \left\{\, (0, \Delta^{+}, \Delta^{+}), (0, 0, \Delta^{+}), (0, \Delta^{+}, 0) \,\right\}$$

$$i = 4 \quad \mathcal{A}(\pi(\Delta^{+}, 0, \Delta, \Delta, \Delta^{+})) = \left\{\, (\Delta^{+}, 0, 0), (\Delta^{+}, 0, \Delta^{+}) \,\right\}$$

  and then for any $0 \leq i \leq 4$, $\mathcal{A}\left(\pi(\mathrm{sh}^i((\Delta^{+}, \Delta^{+}, 0, \Delta, \Delta)))\right) \not\subseteq \left\{\, (0, \Delta^{+}, \Delta^{+}) \,\right\}$;

- $(0, \Delta^{+}, \Delta^{+}) \not\preccurlyeq (\Delta, \Delta^{+}, 0, 0, \Delta^{+})$, it is sufficient to note that it is impossible to find in $(\Delta, \Delta^{+}, 0, 0, \Delta^{+})$ three consecutive components such that first is zero and the others are different from zero.

Remark 3.3.11. Proposition 3.3.9 - (b) proves that $\preccurlyeq$ is a reflexive relation. Unfortunately, it is not transitive in fact $(\Delta^{+}, 0, \Delta^{+}) \preccurlyeq (\Delta^{+}, \Delta^{+}, 0)$ and $(\Delta^{+}, \Delta^{+}, 0) \preccurlyeq (\Delta^{+}, \Delta^{+}, 0, 0, \Delta^{+})$ but $(\Delta^{+}, 0, \Delta^{+}) \not\preccurlyeq (\Delta^{+}, \Delta^{+}, 0, 0, \Delta^{+})$.

3.4 Known strict root bounds

The goal of this section is to show that many known lower bounds are actually strict root bounds. We proceed as follows. We first provide a list of well-known bounds. We then give a "classical" statement for each. Finally, for each we provide a strict root function such that the bound is nothing else that the associated root bound (or a special case), we prove its properties and show the link between the two definitions. Observe that in the case of the four Boston's bounds here analyzed, we do not limit ourselves to reprove them but we generalize them.

We begin with listing the bounds considered in this subsection, citing both their classical statement and their new interpretation in our setting,

- the BCH bound: Theorem 3.4.1 and Corollary 3.4.10,
- the Hartmann-Tzeng (HT) bound: Theorem 3.4.2 and Corollary 3.4.15,
- Boston's bound I, Theorem 3.4.3 and Corollary 3.4.23,
- Boston's bound II, Theorem 3.4.4 and Corollary 3.4.27,
- Boston's bound III, Theorem 3.4.5 and Corollary 3.4.30,
- Boston's bound IV, Theorem 3.4.6 and Corollary 3.4.33,
- the Betti-Sala (BS) bound, Theorem 3.4.7 and Corollary 3.4.36,

3.4.1 "Classical statement" of bounds

We now give "classical statements" for the listed bounds.
The following theorem was first presented in [BRC60] (also [Chi72], [Hoc59]).

Theorem 3.4.1 (BCH bound). *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$ with generator polynomial g. Suppose that there exist $i,\ell\ \in\{0,\dots n-1\}$ such that:*

$$g(\alpha^{i+j})=0,\quad 0\le j\le \ell-1\,.$$

Then:

$$d\ge \ell+1.$$

The following theorem was first presented in [HT72], but the following version is an improvement due to Roos [Roo82].

Theorem 3.4.2 (Hartmann-Tzeng bound). *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$ with generator polynomial g. Suppose that there exist $i_0, \ell, s, r \in \mathbb{N}$ s.t. $(r,n) \leq \ell$ and*

$$g(\alpha^{i_0+i+jr}) = 0, \quad 0 \leq i \leq \ell - 1,\ 0 \leq j \leq s-1\,.$$

Then

$$d \geq \ell + s.$$

The following four theorems were first presented in [Bos01].

Theorem 3.4.3 (Boston bound I). *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. If $3 \nmid n$ and $\{0,1,3,4\} \subseteq S$, then*

$$d \geq 4.$$

Theorem 3.4.4 (Boston bound II). *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. If $\{0,1,3,5\} \subseteq S$, then*

$$d \geq 4.$$

Theorem 3.4.5 (Boston bound III). *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. If $3 \nmid n$ and $\{0,1,3,4,6\} \subseteq S$, then*

$$d \geq 5\,.$$

Theorem 3.4.6 (Boston bound IV). *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. If $4 \nmid n$ and $\{0,1,2,4,5,6,8\} \subseteq S$, then*

$$d \geq 6\,.$$

The following theorem was first presented in [BS05] and [Bet05].

Theorem 3.4.7 (Betti-Sala bound). *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. Suppose that there are $m, \ell \in \mathbb{N}$, $m, \ell \geq 1$ and $i_0 \in \{0,\dots,n-1\}$ such that:*

a) $(i_0 + j)_n \in S$, $j = 0, \dots, m\ell - 1$, *b)* $(i_0 + j)_n \in S$, $j = (m+h)\ell + 1, \dots, (m+h)\ell + \ell - 1$, $0 \leq h \leq m$,

or also such that

c) $(i_0 + j)_n \in S$, $j = h\ell, \dots, h\ell + \ell - 2$, $0 \leq h \leq m$,

d) $(i_0 + j)_n \in S$, $j = (m+1)\ell, \dots, (2m+1)\ell - 1$.

Then:

$$d \geq m\ell + \ell\,.$$

3.4.2 Our interpretation of the BCH bound

Definition 3.4.8. *Let f_{BCH} be the following map $f_{\mathrm{BCH}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\mathrm{BCH}}(n, S) = \max\{i \ \in \ \mathbb{N} \mid (0)^i \preccurlyeq R(n, S)\}\,.$$

Theorem 3.4.9. *Map f_{BCH} is a strict root function.*

Proof. Suppose that $f_{\mathrm{BCH}}(n, S) = \ell + 1$, so that $(0)^\ell \preccurlyeq R(n, S)$. It is enough to show that for any $\mathbf{v} \in \mathcal{A}(n, S)$, we have that $\mathrm{rk}(M(\mathbf{v})) \geq \ell + 1$.
Since $(0)^\ell \preccurlyeq R(n, S)$, any $\mathbf{v} \in \mathcal{A}(n, S)$ contains a block of the form $(0)^j$, with $j \geq \ell$, and by Lemma 3.2.9, we can suppose it lies at the beginning of $\mathbf{v}$. Then

$$\mathbf{v} = (\overbrace{0, 0, \dots, 0}^{j}, \Delta^+, *, \dots, *), \qquad j \geq \ell\,.$$

By Lemma 3.2.14 we have that $\mathrm{rk}(M(\mathbf{v})) \geq j + 1 \geq \ell + 1$. □

The following corollary is then obvious.

Corollary 3.4.10. *The BCH bound is a strict root bound and it is the bound associated to f_{BCH}.*

In particular we have reproved Theorem 3.4.1.

Remark 3.4.11. Many known bounds do not provide explicitly a bound from S, but they give patterns to be searched for in S and then the actual bound to be taken is the largest bound guaranteed by these patterns. For example the classical formulation of the BCH bound does not say explicitly that $d \geq \ell + 1$, with ℓ the largest for which one can apply Theorem 3.4.1. However, this is done by everyone that actually computes the BCH bound for a code. From now on, we will ignore this small formal problem in order not to overburden our notation.

Remark 3.4.12. Both the BCH bound and the HT bound are well-known, so that writing another proof for them may appear superfluous. The reader should note that we do more than reproving them: we prove that they are strict root functions. This has a number of implications. For example, the optimal root bound $\mathsf{f}^{\mathcal{D}}$ will be automatically sharper and tighter than both. As a consequence, they cannot be tight on codes where the rank over $\mathcal{U}$ of $M(\mathbf{u})$ is strictly lower than the rank over the actual field, and hence their tightness will strongly depend on the field.

3.4.3 Our interpretation of the HT bound

Definition 3.4.13. *For any $r, s, n \in \mathbb{N}$ we denote by $\rho = \rho(r, s, n)$ the quotient of rs divided by n and increased by 1.*
Let f_{HT} be the following map $f_{\mathrm{HT}} : \mathcal{D} \to \mathbb{N}$,

$$f_{\mathrm{HT}}(n, S) \;=\; \max\{i \in \ \mathbb{N} \mid i = \ell + s\}\,,$$

where $\ell, s \in \mathbb{N}$, $\ell, s \geq 1$, are such that there exists $r \in \mathbb{N}$, $(r, n) \leq \ell$, for which

$$((0^{\ell})(\Delta^{r-\ell}))^{s} \preccurlyeq R(n, S)^{\rho}. \tag{3.15}$$

Note that $((0^{\ell})(\Delta^{r-\ell}))^{s} \preccurlyeq R(n, S)^{\rho'}$, with $\rho' \geq \rho$, then $((0^{\ell})(\Delta^{r-\ell}))^{s} \preccurlyeq R(n, S)^{\rho}$. We state a theorem postponing its proof.

Theorem 3.4.14. *Map f_{HT} is a strict root function.*

Corollary 3.4.15. *The HT bound is a strict root bound and it is the bound associated to f_{HT}.*

Proof. The first assumption of Theorem 3.4.2 states that $R(n, S)^{\rho}$ contains a block of length $m = rs$ of the form

$$((0^{\ell})(\Delta^{r-\ell}))^{s}\,.$$

□

We have thus reproved Theorem 3.4.2.

The proof of Theorem 3.4.14 requires a few definitions and lemmas.

Definition 3.4.16. *Let $\mathbf{v} \in (\mathcal{U} \setminus \{\Delta\})^{n}$, $\mathbf{v} \neq \mathbf{0}$, and let $\rho \in \mathbb{N}$. Let $1 \leq i \leq n$. We say that i is the `primary pivot` of $\mathbf{v}$ if $\mathbf{v}[i]$ is the first $\boldsymbol{\Delta}^{+}$ that occurs in $\mathbf{v}$, i.e.*

$$i = \min\{h \mid \mathbf{v}[h] = \boldsymbol{\Delta}^{+}\}\,.$$

Lemma 3.4.17. *Let $n, r, s, \ell \in \mathbb{N}$ such that $(r, n) \leq \ell$. Then for any i in $\{0, \ldots, n-1\}$ there are $k \in \mathbb{N}$ and $0 \leq t \leq \ell - 1$ such that*

$$i \equiv (s+k)r + t \mod (n)\,.$$

Proof. Given $i \in \{0, \ldots, n-1\}$, let $\lambda = (r, n)$. By hypothesis $\lambda \leq \ell$. Let t be such that:

$$i \equiv t \mod (\lambda), \qquad 0 \leq t \leq \ell - 1\,.$$

We have that $\lambda \mid i - t$. In correspondence of this t:

$$\begin{aligned} i \equiv (s+k)r + t \mod (n) &\iff \\ i - t \equiv (s+k)r \mod (n) &\iff \frac{i-t}{\lambda} \equiv (s+k)\frac{r}{\lambda} \mod \left(\frac{n}{\lambda}\right). \end{aligned}$$

By defining $y = s + k$ we obtain

$$\frac{i-t}{\lambda} \equiv y\frac{r}{\lambda} \mod \left(\frac{n}{\lambda}\right).$$

The equation above has a solution y_0, since $\left(\frac{r}{\lambda}, \frac{n}{\lambda}\right) = 1$. If we define $k = y_0 - s$, we have found k and t satisfying our required congruence. □

Note that in the previous lemma $0 \leq i \leq n-1$ and $0 \leq t \leq \ell - 1$, while in the next lemma $1 \leq j \leq n$ and $1 \leq t \leq \ell$.

Lemma 3.4.18. *Let $n, \ell, r, s \in \mathbb{N}$ and let $\mathbf{v} \in (\mathcal{U} \setminus \{\Delta\})^n$, $\mathbf{v} \neq 0$ such that $(n, r) \leq \ell$ and $B = ((0)^\ell(\Delta)^{r-\ell})^s \preccurlyeq \mathbf{v}^\rho$. Then there are $i \in \{1, \ldots n\}$, $k \in \mathbb{N}$ and $t \in \{1, \ldots, \ell\}$, with the following properties:*

1. $\mathbf{v}[i] = \Delta^{\!\!+}$,
2. $i \equiv (s+k)r + t \mod (n)$,
3. $\mathbf{v}[i'] = 0$, *for any i' s.t.*

$$i' \equiv (s+k')r + j \mod (n)\,,$$

where $k' \in \{0, \ldots, k-1\}$ and $j \in \{1, \ldots, \ell\}$.

Proof. It follows directly from Lemma 3.4.17, once we increase by 1 both i and ℓ. □

Definition 3.4.19. *Let us adopt the same notation as in Lemma 3.4.18. We say that i is the* `secondary pivot` *of $\mathbf{v}$ with respect to block B.*

We are ready for the proof of Theorem 3.4.14.

Proof. (Theorem 3.4.14)
Given $(n, S) \in \mathcal{D}$, by definition of f_{HT}, there are ℓ, s, r, i_0 such that $f_{\mathrm{HT}}(n, S) = \ell + s$ and ℓ, s, r, i_0 satisfy the assumptions of Theorem 3.4.2. Given $\mathbf{v} \in \mathcal{A}(n, S)$, it is enough to show that the singleton procedure is successful for $\ell + s$ rows of the matrix $M(\mathbf{v})$. By Lemma 3.2.9 we can suppose that $i_0 = 0$.
Let j be the primary pivot of $\mathbf{v}$. If $j > rs$ then $(0)^{rs} \preccurlyeq \mathbf{v}$ and Theorem 3.4.9 ensures that $\mathrm{rk}(M(\mathbf{v})) \geq rs + 1 \geq \ell + s$ and we have finished. So we may suppose that $j \leq rs$.
Let i be the secondary pivot of $\mathbf{v}$ w. r. t. block $((0)^{\ell}(\Delta)^{r-\ell})^{s}$. Observe that i is such that $\mathbf{v}[i - zr] = 0$, for any $z = 1, \dots, s$. In other words, $\mathbf{v}$ is as follows:

$$\begin{array}{cccccccccccccc}
 & 1 & & & j & & & & & & m=sr & & i & \\
 & \downarrow & & & \downarrow & & & & & & \downarrow & & \downarrow & \\
\mathbf{v} = & 0 & \dots & 0 & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta\ \dots & \Delta & \dots & \Delta^{+} & \dots
\end{array}$$

Note that i and j can coincide. By hypothesis:

$$\mathbf{v}^{\rho} = \overbrace{\overbrace{0, \dots, 0}^{\ell}\overbrace{\Delta, \dots, \Delta}^{r-\ell}, \dots, \overbrace{0, \dots, 0}^{\ell}, \overbrace{\Delta, \dots, \Delta}^{r-\ell}}^{m=sr}, \Delta, \dots \quad ,$$

where Δ denotes either Δ^{+} or 0 (with abuse of notation). We have to choose $\ell + s$ rows of matrix $M(\mathbf{v})$ and to apply the singleton procedure. We start with the first ℓ rows:

$$\begin{array}{ccccccccccccccc}
 & & & j & & & & & & & & & & & \\
 & & & \downarrow & & & & & & & & & & & \\
0 & \dots & 0 & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & & & \\
\Delta & 0 & \dots & 0 & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & & \\
\vdots & & \vdots & & & & & & & \vdots & & \vdots & & & \\
\Delta & \dots & \Delta & 0 & \dots & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots &
\end{array}$$

We now add the $(j - m - 1 + zr)_n + 1$-th row of $M(\mathbf{v})$, for all $z = 1, 2, \dots, s$, thus obtaining an $(\ell + s) \times n$ matrix T, as follows

$$T = \left(\begin{array}{cccccccccccccccc}
0 & \dots & 0 & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & & & & \\
\Delta & 0 & \dots & 0 & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & & & \\
\vdots & & \vdots & & & & & & & \vdots & & \vdots & & & & \\
\Delta & \dots & \Delta & 0 & \dots & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & & \\
\Delta & \dots & \Delta & 0 & \dots & 0 & \Delta & \dots & & & & & & & & \\
\Delta & \dots & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta & 0 & \dots & 0 & \Delta & \dots & & \\
 & \vdots & & & & & & & & & & & & \vdots & & \\
\Delta & \dots & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta & 0 & \dots & 0 & \Delta & \dots & & \\
 & & & \uparrow & & \uparrow & & & & & & & & & & \\
 & & & j & & j+\ell-1 & & & & & & & & & &
\end{array}\right)$$

Observe that the rows from row $\ell + 1$ to row n have a zero-block of length ℓ exactly from the j-th position and the $j + \ell - 1$-th position (see Remark 3.4.20). We have so obtained a sub-matrix T of $M(\mathbf{v})$, for which the first ℓ rows can be obviously erased by the singleton procedure. After this first application of the procedure, we are left with a matrix T' composed of the last s rows of T, as follows:

$$T' = \begin{pmatrix} \Delta \dots \Delta \; 0 \dots 0 \; \Delta \dots \Delta \dots \Delta^{+} \; \Delta \dots \\ \Delta \dots \Delta \; 0 \dots 0 \; \Delta \dots \Delta \;\; 0 \;\; \dots \;\; 0 \;\; \Delta \;\dots \\ \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \Delta \dots \Delta \; 0 \dots 0 \; \Delta \dots \Delta \;\; 0 \;\; \dots \;\; 0 \;\; \Delta \;\dots \end{pmatrix}$$

By construction, we note that T' has the property $T'[a+1,h] = T'[a,h-r]$, because each row is obtained by an r-th shift of the previous one.

For $1 \leq z \leq s$ let $i'_z = (i+j-m-2+zr)_n + 1$, that is, i_z is the secondary pivot in the z-th row of T'. We know that $T'[1,i'_1] = \Delta^{+}$ and this is sufficient to establish that the i'_1-th column is a singleton, since:

$$T'[z,i'_1] = T'[1,i'_1 + (1-z)r] = 0, \quad z = 2,\dots,s\,.$$

Then we erase the first row, and repeat the same for the second one, using as singleton the i'_2-th column:

$$\begin{aligned} &T'[2,i'_2] = T'[2,i'_1 + r] = T'[1,i'_1] = \Delta^{+} \\ &T'[z,i'_2] = T'[1,i'_1 + (2-z)r] = 0,\ z = 3,\dots,s\,. \end{aligned}$$

In this way, for any z-th row of T' from 1 to s we have a singleton at position i'_z and that means that the singleton procedure is successful for matrix T', implying that the procedure is successful also for T, as claimed. □

Remark 3.4.20. We want to comment the previous proof, highlighting the relation between the rows to be checked by the singleton procedure to prove a bound and the pattern of blocks that defines the bound. The HT bound is a generalization of the BCH bound in the sense that if in $R(n,S_C)$ there are s blocks of type $(0)^{\ell}$, then the BCH bound can be increased by $s-1$. However, this is true only if there is an $r \geq 1$ such that any two consecutive blocks are at distance r and if $\gcd(r,n) \leq \ell$.

To prove the bound we need to choose $\ell + s$ rows in $M(\mathbf{v})$, for $\mathbf{v} \in \mathcal{A}(n,S)$, on which the singleton procedure is successful. It would be obvious to use the first $\ell + 1$ rows, since they guarantee the BCH bound, but we take only the first ℓ rows. The problem is that we need other rows and they have to be chosen in order not to hamper the check for the remaining ones. The primary pivot is the Δ^{+} needed to delete the first rows. So, if we choose only the first ℓ rows, then we can find s rows in such a way that they have a 0 under the primary pivot and hence the deletion of the first ℓ rows will not be hampered by the new rows. To find our missing s rows, we need the existence of r. At this stage we have s rows, but we need at least a Δ^{+} to delete them and this is exactly the role of our secondary pivot. Its existence is guaranteed by the second condition, i.e. $\gcd(r,n) \leq \ell$. To grasp this, we propose to apply the singleton procedure to $R(n,S) = (0,0,\Delta,0,0,\Delta)$. It satisfies the hypothesis of Theorem 3.4.14, except for $\gcd(r,n) \leq \ell$, and it would give $d \geq 5$ with $n = 6, r = 3, \ell = 2$, which is easily seen to be impossible. The point is that no secondary pivot can be found, as it is clear in the following matrix, where we have removed the first two rows ($\ell = 2$):

$$\begin{pmatrix} 0 & 0 & \Delta^{+} & 0 & 0 & \Delta^{+} \\ \Delta^{+} & 0 & 0 & \Delta^{+} & 0 & 0 \\ 0 & \Delta^{+} & 0 & 0\Delta^{+} & 0 & \\ 0 & \Delta^{+} & 0 & 0 & \Delta^{+} & 0 \end{pmatrix}$$

3.4.4 Our interpretation of Boston's bound I

Definition 3.4.21. *Let f_{B1} be the following map $f_{\mathrm{B1}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\mathrm{B1}}(n,S) = \begin{cases} 4, & \textit{if } (0,0,\Delta,0,0) \preccurlyeq R(n,s) \textit{and } 3 \nmid n, \\ 1, & \textit{otherwise.} \end{cases}$$

Theorem 3.4.22. *Map f_{B1} is a strict root function.*

Proof. It is a special case of map f_{HT}, with $\ell = 2$ and $s = 2$. □

The following corollary is then obvious.

Corollary 3.4.23. *Boston's bound I is a strict root bound and it is implied by the bound associated to f_{B1}.*

In particular we have reproved Theorem 3.4.3.

Remark 3.4.24. In our statement Corollary 3.4.23 we say "it is implied by", where we mean that we have replaced condition $\{0,1,3,4\} \preccurlyeq S$ by the more general condition $\{i, i+1, i+3, i+4\} \subseteq S$, where i is any integer such that $0 \leq i \leq n-4$. Actually, we obtain this kind of generalization for all these Boston's bounds and this is an interesting consequence of Lemma 3.2.9.

3.4.5 Our interpretation of Boston's bound II

Definition 3.4.25. *Let f_{B2} be the following map $f_{\mathrm{B2}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\mathrm{B2}}(n,S) = \begin{cases} 4, & \textit{if } (0,0,\Delta,0,\Delta,0) \preccurlyeq R(n,s), \\ 1, & \textit{otherwise.} \end{cases}$$

Theorem 3.4.26. *Map f_{B2} is a strict root function.*

Proof. It is enough to show that for any $\mathbf{v} \neq 0$ such that $(0,0,\Delta,0,\Delta,0)$ is contained in $\mathbf{v}$, we have $\mathrm{rk}(M(\mathbf{v})) \geq 4$. We can suppose by Lemma 3.2.9 that our block lies at the beginning of $\mathbf{v}$.
We consider two cases, which altogether cover all possibilities, as follows:

a. $(0,0,0,0,\Delta,0) \preccurlyeq \mathbf{v}$,

b. $(0,0,\Delta^{\!\!+},0,\Delta,0) \preccurlyeq \mathbf{v}$.

Case a
Since $(0,0,0,0,\Delta,0) \preccurlyeq \mathbf{v}$, we have $(0)^4 \preccurlyeq \mathbf{v}$ and hence this is a special case of f_{BCH}, which ensures $\mathrm{rk}(M(\mathbf{v})) \geq 5$.

Case b
Let A_4 be the sub-matrix of $M(\mathbf{v})$ formed by rows $\{1,2,3,n\}$.

$$A_4 = \begin{pmatrix} 0 & 0 & \Delta^{\!\!+} & 0 & \Delta & 0 & \Delta & \Delta & \dots \\ \Delta & 0 & 0 & \Delta^{\!\!+} & 0 & \Delta & 0 & \Delta & \dots \\ \Delta & \Delta & 0 & 0 & \Delta^{\!\!+} & 0 & \Delta & \Delta & \dots \\ 0 & \Delta^{\!\!+} & 0 & \Delta & 0 & \Delta & \Delta & \Delta & \dots \end{pmatrix}$$

The singleton procedure is successful for A_4, since erasing in order the following rows $\{1,3,4\}$ yields $A_1 = (\Delta,\Delta^{\!\!+},\Delta,0,\Delta,\dots)$. □

The following corollary is then obvious.

Corollary 3.4.27. *Boston's bound II is a strict root bound and it is a special case of the bound associated to f_{B2}.*

In particular we have reproved Theorem 3.4.4 (see Remark 3.4.24).

3.4.6 Our interpretation of Boston's bound III

Definition 3.4.28. *Let f_{B3} be the following map $f_{\mathrm{B3}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\mathrm{B3}}(n,S) = \begin{cases} 5, & \textit{if } (0,0,\Delta,0,0,\Delta,0) \preccurlyeq R(n,s) \textit{ and } 3 \nmid n, \\ 1, & \textit{otherwise.} \end{cases}$$

Theorem 3.4.29. *Map f_{B3} is a strict root function.*

Proof. It is enough to show that for any $\mathbf{v} \neq 0$ s.t. $(0,0,\Delta,0,0,\Delta,0) \preccurlyeq \mathbf{v}$ and $3 \nmid n$, we have $\mathrm{rk}(M(\mathbf{v})) \geq 5$. We can suppose by Lemma 3.2.9 that our block is at the beginning of $\mathbf{v}$.
We consider four cases, which altogether cover all possibilities, as follows:

a. $(0,0,0,0,0,\Delta,0) \preccurlyeq \mathbf{v}$,

b. $(0,0,\Delta^{+},0,0,0,0) \preccurlyeq \mathbf{v}$,

c. $(0,0,\Delta^{+},0,0,\Delta^{+},0,0) \preccurlyeq \mathbf{v}$,

d. $(0,0,\Delta^{+},0,0,\Delta^{+},0,\Delta^{+}) \preccurlyeq \mathbf{v}$.

Case a
Since $(0,0,0,0,0,\Delta,0) \preccurlyeq \mathbf{v}$, we have $(0)^5 \preccurlyeq \mathbf{v}$ and hence this is a special case of f_{BCH}, which ensures $\mathrm{rk}(M(\mathbf{v})) \geq 6$.

Case b
Since $(0,0,\Delta^{+},0,0,0,0) \preccurlyeq \mathbf{v}$, we have $(0)^4 \preccurlyeq \mathbf{v}$ and hence this is a special case of f_{BCH}, which ensures $\mathrm{rk}(M(\mathbf{v})) \geq 5$.

Case c
Since $(0,0,\Delta^{+},0,0,\Delta^{+},0,0) \preccurlyeq \mathbf{v}$, we have $((0)^2(\Delta)^1)^3 \preccurlyeq \mathbf{v}$ and hence this is a special case of f_{HT} with $l=2$ and $s=3$, which ensures $\mathrm{rk}(M(\mathbf{v})) \geq 2+3=5$ (since $s=3 \nmid n$).

Case d
Let A_5 be the sub-matrix of $M(\mathbf{v})$ formed by rows $\{1,2,3,n,n-1\}$.

$$A_5 = \begin{pmatrix} 0 & 0 & \Delta^{+} & 0 & 0 & \Delta & 0 & \Delta^{+} & \Delta & \dots \\ \Delta & 0 & 0 & \Delta^{+} & 0 & 0 & \Delta & 0 & \Delta & \dots \\ \Delta & \Delta & 0 & 0 & \Delta^{+} & 0 & 0 & \Delta & \Delta & \dots \\ 0 & \Delta^{+} & 0 & 0 & \Delta & 0 & \Delta^{+} & \Delta & \Delta & \dots \\ \Delta^{+} & 0 & 0 & \Delta & 0 & \Delta^{+} & \Delta & \Delta & \Delta & \dots \end{pmatrix}$$

The singleton procedure is successful for A_5, since erasing in order the following rows $\{1,5,2,4\}$ yields $A_1 = (\Delta,\Delta,\Delta^{+},0,\Delta,\Delta,\ldots)$. □

The following corollary is then obvious.

Corollary 3.4.30. *Boston's bound III is a strict root bound and it is a special case of the bound associated to f_{B3}.*

In particular we have reproved Theorem 3.4.5 (see Remark 3.4.24).

3.4.7 Our interpretation of Boston's bound IV

Definition 3.4.31. *Let f_{B4} be the following map $f_{\mathrm{B4}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\mathrm{B4}}(n, S) = \begin{cases} 6, & \textit{if } (0,0,0,\Delta,0,0,0,\Delta,0) \preccurlyeq R(n,s) \textit{ and } 4 \nmid n, \\ 1, & \textit{otherwise.} \end{cases}$$

Theorem 3.4.32. *Map f_{B4} is a strict root function.*

Proof. It is enough to show that for any $\mathbf{v} \neq 0$ s.t. $(0,0,0,\Delta,0,0,0,\Delta,0) \preccurlyeq \mathbf{v}$ and $4 \nmid n$, we have $\mathrm{rk}(M(\mathbf{v})) \geq 6$. We can suppose by Lemma 3.2.9 that our block is at the beginning of $\mathbf{v}$.
We consider four cases, which altogether cover all possibilities, as follows:

a. $(0,0,0,0,0,0,0,\Delta,0,\Delta,\Delta) \preccurlyeq \mathbf{v}$,

b. $(0,0,0,\Delta^{\!\!+},0,0,0,\Delta,0,0,0) \preccurlyeq \mathbf{v}$,

c. $(0,0,0,\Delta^{\!\!+},0,0,0,\Delta,0,\Delta^{\!\!+},\Delta) \preccurlyeq \mathbf{v}$,

d. $(0,0,0,\Delta^{\!\!+},0,0,0,\Delta,0,0,\Delta^{\!\!+}) \preccurlyeq \mathbf{v}$.

Case a
Since $(0,0,0,0,0,0,0,\Delta,0,\Delta,\Delta) \preccurlyeq \mathbf{v}$, we have $(0)^7 \preccurlyeq \mathbf{v}$ and hence this is a special case of f_{BCH}, which ensures $\mathrm{rk}(M(\mathbf{v})) \geq 8$.

Case b
Since $(0,0,0,\Delta^{\!\!+},0,0,0,\Delta,0,0,0) \preccurlyeq \mathbf{v}$, we have $((0)^3(\Delta)^1)^3 \preccurlyeq \mathbf{v}$ and hence this is a special case of f_{HT}, with $l = 3$ and $s = 3$, which will give exactly $\mathrm{rk}(M(\mathbf{v})) \geq 3+3 = 6$ (since $s = 4 \nmid n$).
Case c
Let A_6 be the sub-matrix of $M(\mathbf{v})$ formed by rows $\{1,2,3,4,n,n-1\}$.

$$A_6 = \begin{pmatrix} 0 & 0 & 0 & \Delta^{\!\!+} & 0 & 0 & 0 & \Delta & 0 & \Delta^{\!\!+} & \Delta & \ldots \\ \Delta & 0 & 0 & 0 & \Delta^{\!\!+} & 0 & 0 & 0 & \Delta & 0 & \Delta & \ldots \\ \Delta & \Delta & 0 & 0 & 0 & \Delta^{\!\!+} & 0 & 0 & 0 & \Delta & \Delta & \ldots \\ \Delta & \Delta & \Delta & 0 & 0 & 0 & \Delta^{\!\!+} & 0 & 0 & 0 & \Delta & \ldots \\ 0 & 0 & \Delta^{\!\!+} & 0 & 0 & 0 & \Delta & 0 & \Delta^{\!\!+} & \Delta & \Delta & \ldots \\ 0 & \Delta^{\!\!+} & 0 & 0 & 0 & \Delta & 0 & \Delta^{\!\!+} & \Delta & \Delta & \Delta & \ldots \end{pmatrix}$$

The singleton procedure is successful for A_6, since erasing in order the following rows $\{1,2,6,3,5\}$ yields $A_1 = (\Delta,\Delta,\Delta,\Delta^{\!\!+},0,\Delta,\ldots)$.

Case d
Let A_6 be the sub-matrix of $M(\mathbf{v})$ formed by rows $\{1, 2, 3, 4, n, n-1\}$.

$$A_6 = \begin{pmatrix} 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & 0 & \Delta & 0 & 0 & \Delta^{\!+} & \Delta & \dots \\ \Delta & 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & 0 & \Delta & 0 & 0 & \Delta & \dots \\ \Delta & \Delta & 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & 0 & \Delta & 0 & \Delta & \dots \\ \Delta & \Delta & \Delta & 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & 0 & \Delta & \Delta & \dots \\ 0 & 0 & \Delta^{\!+} & 0 & 0 & 0 & \Delta & 0 & 0 & \Delta^{\!+} & \Delta & \Delta & \dots \\ 0 & \Delta^{\!+} & 0 & 0 & 0 & \Delta & 0 & 0 & \Delta^{\!+} & \Delta & \Delta & \Delta & \dots \end{pmatrix}$$

The singleton procedure is successful for A_6, since erasing in order the following rows $\{1, 2, 6, 3, 5\}$ yields $A_1 = (\Delta, \Delta, \Delta, \Delta^{\!+}, 0, \Delta, \dots)$. □

The following corollary is then obvious.

Corollary 3.4.33. *Boston's bound IV is a strict root bound and it is a special case of the bound associated to f_{B4}.*

In particular we have reproved Theorem 3.4.6 (see Remark 3.4.24).

3.4.8 Our interpretation of the BS bound

Definition 3.4.34. *Let f_{BS} be the following map $f_{\mathrm{BS}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\mathrm{BS}}(n, S) = \max\{i \in \mathbb{N} \mid i = m\ell + \ell\},$$

where m and ℓ are s.t. either

$$((0)^{\ell})^{m}((\Delta)^{1}(0)^{\ell-1})^{m+1} \preccurlyeq R(n, S), \tag{3.16}$$

or

$$((0)^{\ell-1}(\Delta)^{1})^{m+1}((0)^{\ell})^{m} \preccurlyeq R(n, S). \tag{3.17}$$

Theorem 3.4.35. *Map f_{BS} is a strict root function.*

Proof. We briefly summarize and adapt arguments from [BS06].

Note that (3.17) is the reflection of (3.16), so that by Lemma 3.2.13 it is sufficient to consider (3.16).

As usual, it is enough to show that for any $\mathbf{v} \neq 0$ s.t.

$$((0)^{\ell})^{m}((\Delta)^{1}(0)^{\ell-1})^{m+1} \preccurlyeq \mathbf{v},$$

we have $\mathrm{rk}(M(\mathbf{v})) \geq m\ell + \ell$. Let T be the sub-matrix of $M(\mathbf{v})$ formed by the first $m\ell + \ell$ rows. We want to show that the singleton procedure is successful for T. We can suppose that the block is at the beginning of $\mathbf{v}$. We have two cases: either $\mathbf{u}[m\ell{+}1] = 0$ or $\mathbf{u}[m\ell{+}1] = \Delta^{+}$. In the first case we have $(0)^{j_0} \preccurlyeq \mathbf{v}$, for some $j_0 \geq ml{+}l$ and so we may apply Lemma 3.2.14, ensuring that $\mathrm{rk}(M(\mathbf{u})) \geq j_0 + 1 > m\ell + \ell$.

In the second case $\mathbf{u}$ starts with block $(0)^{m\ell}(\Delta^{+})^1(0)^{\ell-1}((\Delta)^1(0)^{\ell-1})^m$. We apply the singleton procedure to T using as singleton $T[(m{+}i)\ell{+}j]$, with j decreasing from ℓ to 1 and, for any fixed j, i increasing from 0 to m. It is clearly sufficient to verify that at each step column $T[(m+i)\ell+j]$ is a singleton. Let us consider a generic step of the procedure, with some i and j. By circularity we have

$$T[i\ell + j, (m+i)\ell + j] = T[i\ell + j - 1, (m+i)\ell + j - 1] = \dots$$

$$\dots = T[1, m\ell + 1] = \mathbf{u}[m\ell + 1] = \Delta^{+} .$$

Suppose that there exists another $s \in \{2, \dots, m\ell + \ell\}$ s.t. $T[s, (m+i)\ell + j] = \Delta^{+}$, which means $\mathbf{u}[(m+i)\ell + j - s + 1] = \Delta^{+}$. By the assumptions on the structure of $\mathbf{u}$, we get $\ell(m+i) + j - s = (m+h)\ell$, $h \geq 0$, which implies $s = (i-h)\ell + j$, for $h \geq 0$. If $h = 0$, we have $s = i\ell + j$ as required. If $h \geq 1$, we have $s = i'\ell + j$, with $i' < i$. But the $s-$th row has already been erased in some previous step of the procedure. We then conclude that $T[\ell(m+i) + j]$ is a singleton. □

Corollary 3.4.36. *Bound BS is a strict root bound and it is the bound associated to the strict root function f_{BS}.*

Proof. In case $i_0{+}m\ell{-}1 \leq n{-}1$, condition a) of Theorem 3.4.7 states that S contains $m\ell$ consecutive integers. In case $i_0 + m\ell - 1 > n - 1$, condition a) means that there are two blocks of consecutive integers in S: one from i_0 to $n-1$ and one from 0 to $i_0{+}m\ell{-}1{-}n$, so that we can still view this case as describing a block of "consecutive" integers in S (the "large block"). On the other hand, condition b) of Theorem 3.4.7 implies that for any h there is a block of $\ell - 1$ "consecutive" integers in S (a "small block"), that between two small blocks there is an integer $i' = i_0 + ((m+h)\ell)_n$ s. t. we do not know if i' is in S, and that between the large block and the first small block, there is an integer $i'' = i_0 + (m\ell)_n$ s. t. we do not know if i'' is in S. In other words, the assumptions a) and b) in Theorem 3.4.7 are equivalent to saying that $R(n, S)$ "contains" a block:

$$\overbrace{\overbrace{0,\dots,0}^{\ell},\dots,\overbrace{0,\dots,0}^{\ell}}^{m\ell},\overbrace{\overbrace{\Delta,0,\dots,0}^{\ell},\dots,\overbrace{\Delta,0,\dots,0}^{\ell}}^{(m+1)\ell}$$

Similarly for c) and d) (but using Lemma 3.2.13). □

3.5 Known root bounds which are not strict

All bounds presented in Subsection 3.4 are strict root bounds. This subsection deals with two other root bounds, i.e. the Roos bound and Boston's bound V. We are able to show that they are root bounds, but not strict root bounds. To obtain this result we show that *it is not possible to prove that they are strict root bounds using the singleton procedure.* Since the failure of the singleton procedure implies the linear dependence of the rows (see Theorem 3.2.18), we conclude that indeed they are not strict root bounds.

The following result was first presented in [Roo83] and we do not give an alternative proof.

Theorem 3.5.1 (Roos bound)**.** *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$ with generator polynomial g. Let $r \in \mathbb{N}$ s.t.*[1] *$2 \leq r \leq n-1$ and $(r,n)=1$. Let $\ell \in \mathbb{N}$, $2 \leq \ell \leq n-1$.*
Let $\bar{S}$ be a set of $\bar{s}$ consecutive natural numbers: $\bar{S} := \{k, k+1, \dots, k+\bar{s}-1\}$. Let $S' \subseteq \bar{S}$, $|S'| = s$, s.t.[2]

$$\bar{s} - s < \ell\,.$$

Suppose that, for an $0 \leq i_0 \leq n-1$, we have

$$g(\alpha^{i_0+i+\sigma r}), \qquad \textit{for } 0 \leq i \leq \ell-1 \textit{ and } \sigma \in S'\,.$$

Then

$$d \geq \ell + s$$

We formalize the Roos bound within our context.

Definition 3.5.2. *Let f_{Roos} be the following map $f_{\mathrm{Roos}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\mathrm{Roos}}(n,S) = \max\{i \in \ \mathbb{N} \mid i = \ell + s\}\,,$$

where ℓ, s are such that there exists $r \in \mathbb{N}$, $(r,n)=1$, and there exist s integers $0 \leq k_1 < k_2 < .. < k_s < \ell + s$, so that:

$$(\Delta)^{rk_1}(0)^{\ell}(\Delta)^{r-\ell}(\Delta)^{r(k_2-k_1-1)}(0)^{\ell}(\Delta)^{r-\ell}\cdots(\Delta)^{r(k_s-k_{s-1}-1)}(0)^{\ell}(\Delta)^{r-\ell} \preccurlyeq R(n,S)^{\rho}. \tag{3.18}$$

where ρ is the remainder of $(k_s+1)r$ in the division by n, increased by 1.

Theorem 3.5.3. *Map f_{Roos} is a root function and the Roos bound is the root bound associated to it.*

[1] *r is s.t. α^r is another primitive n-th root of unity*

[2] *i.e., S' is obtained from S by removing strictly less than ℓ elements.*

Proof. It follows from Theorem 3.5.1, since (3.18) is nothing else but a rewriting of the assumptions of said theorem. □

The following theorem was first presented in [Bos01].

Theorem 3.5.4 (Boston bound V)**.** *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n, k, d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. If $3 \nmid n$ and $\{0, 1, 3, 4, 6, 7\} \subseteq S$, then*

$$d \geq 6\,.$$

We formalize Boston's bound V within our context.

Definition 3.5.5. *Let f_{B5} be the following map $f_{\text{B5}} : \mathcal{D} \to \mathbb{N}$,*

$$f_{\text{B5}}(n, S) = \begin{cases} 6, & \textit{if } R(n, s) = (0, 0, \Delta, 0, 0, \Delta, 0, 0, \Delta, \ldots) \textit{ and } 3 \nmid n, \\ 1, & \textit{otherwise.} \end{cases} \tag{3.19}$$

Theorem 3.5.6. *Map f_{B5} is a root function and Boston's bound V is the associated root bound.*

Proof. It follows from Theorem 3.5.4, since (3.19) is a simple rewriting of the assumptions of said theorem. □

Theorem 3.5.7. *The Boston bound V and the Roos bound are not strict root bounds*

Proof. Let

$$\begin{aligned} \mathbf{v} = & \qquad\qquad (0, 0, \Delta^+, 0, 0, \Delta^+, 0, 0, \Delta^+, \Delta^+, \Delta^+, \Delta^+, \Delta^+), \\ \mathbf{v}' = & \ (0, 0, \Delta^+, \Delta^+, \Delta^+, \Delta^+, 0, 0, \Delta^+, 0, 0, \Delta^+, \Delta^+, \Delta^+, \Delta^+, \Delta^+, \Delta^+, \Delta^+, \Delta^+, \Delta^+)\,, \end{aligned}$$

where $\mathbf{v} \in \mathcal{A}(R(13, S))$, $\mathbf{v}' \in \mathcal{A}(R(20, S'))$,and

$$S = \{0, 1, 3, 4, 6, 7\}, \quad S' = \{0, 1, 6, 7, 9, 10\}\,.$$

If the Boston bound V were a strict root bound, then we would be able to find a 6-row submatrix N in $M(\mathbf{v})$ with $\text{rk}(N) = 6$ on which the singleton procedure is successful (see Theorem 3.2.18). By a computer search, running the singleton procedure on all possible six rows submatrices of $M(\mathbf{v})$ we checked (see Section 9.3) that no such submatrix exists and that actually the rank is 5.

Similarly, if the Roos bound were a strict root bound, then we would be able to find a 5-row submatrix N in $M(\mathbf{v}')$ on which the singleton procedure is successful. By a computer search we checked that no such N exists and that actually the rank of $M(\mathbf{v}')$ is 4 (see Section 9.3). □

Theorem 3.5.8.

$$\mathsf{f}^S < \mathsf{f}$$

Proof. It is clear that $\mathsf{f} \geq \mathsf{f}^S$, we have only to exhibit an (n, S) such that $\mathsf{f}^S(n, S) < \mathsf{f}(n, S)$. From Theorem 3.5.3 we have $f_{\text{Roos}} \in \mathcal{R}$ and then $f_{\text{Roos}} \leq \mathsf{f}$. Let $n = 20$, $S = \{0, 1, 6, 7, 9, 10\}$, from Theorem 3.3.7 and Theorem 3.5.7 we have $\mathsf{f}^S(n, S) \leq 4$ while $\mathsf{f}(n, S) \geq f_{\text{Roos}}(n, S) = 5$, so

$$\mathsf{f}^S(n, S) \leq 4 < 5 = f_{\text{Roos}}(n, S) \leq \mathsf{f}(n, S).$$

□

3.6 Counterexamples to known bounds

The following two theorems were claimed in [Bos01].

Theorem 3.6.1 (Boston A)**.** *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n, k, d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. If $4 \nmid n$ and $\{0, 1, 4, 5, 8\} \subseteq S$, then*

$$d \geq 5\,.$$

Theorem 3.6.2 (Boston B)**.** *Let α be an n-th primitive root of unity over $\mathbb{F}_q$, and let C be an $[n, k, d]$ cyclic code over $\mathbb{F}_q$. Let S be the complete defining set of C w.r.t. α. If $3 \nmid n$ and $\{0, 1, 3, 4, 6, 7, 9\} \subseteq S$, then*

$$d \geq 7\,.$$

Let C be the cyclic code of length 15 over $\mathbb{F}_2$ with complete defining set (w.r.t. to any of the primitive 15-th roots of unity)

$$S_{C,\alpha} = \{0, 1, 2, 4, 5, 8, 10\}\,.$$

Let d be the distance of C. According to Theorem 3.6.1, d should be $d \geq 5$. However, a direct computation (see Section 9.3) shows that $d = 4$, which means:

Theorem 3.6.3. *Theorem 3.6.1 is false.*

Let C be the cyclic code of length 20 over $\mathbb{F}_{11}$ with complete defining set (w.r.t. to any of the primitive 20-th roots of unity) is

$$S_{C,\alpha} = \{0, 1, 3, 4, 5, 6, 7, 9, 11, 13, 15, 17, 19\}\,.$$

Let d be the distance of C. According to Theorem 3.6.2, d should be $d \geq 7$. However, a direct computation (see Section 9.3) shows that $d = 6$, which means:

Theorem 3.6.4. *Theorem 3.6.2 is false.*

Remark 3.6.5. These two statements, along with the other statements discussed in Subsection 3.4, are presented as "corollaries" in [Bos01]. For one of the two statements the author explains that he is still not sure of the result, since it apparently depends on some unfinished computer computations.

3.7 Deducing other bounds

Thanks to our approach, it is easy to deduce new bounds from the bounds presented until now.

By applying Definition 3.2.11 and Lemma 3.2.13 to our generalizations of Boston's bounds II, III and IV, we obtain:

Theorem 3.7.1. *The following functions are strict root functions*

$$f_{\mathrm{B2+}} : \mathcal{D} \to \mathbb{N}, \quad f_{\mathrm{B2+}}(n,S) = \begin{cases} 4, & \textit{if } (0,\Delta,0,\Delta,0,0) \subseteq R(n,S), \\ 1, & \textit{otherwise.} \end{cases}$$

$$f_{\mathrm{B3+}} : \mathcal{D} \to \mathbb{N}, \quad f_{\mathrm{B3+}}(n,S) = \begin{cases} 5, & \textit{if } (0,\Delta,0,0,\Delta,0,0) \subseteq R(n,S) \textit{ and } 3 \nmid n, \\ 1, & \textit{otherwise.} \end{cases}$$

$$f_{\mathrm{B4+}} : \mathcal{D} \to \mathbb{N}, \quad f_{\mathrm{B4+}}(n,S) = \begin{cases} 6, & \textit{if } (0,\Delta,0,0,0,\Delta,0,0,0) \subseteq R(n,S)\,,\ 4 \nmid n, \\ 1, & \textit{otherwise.} \end{cases}$$

We cannot use similar argument with Boston's bound I, the HT bound, the BCH bound and bound A, since their formulation is already symmetric.

Remark 3.7.2. All root bounds analyzed in this section are monotone.

Border bounds

This chapter belongs to a work joint with E. Betti, relates the results contained in the unpublished paper [BS07] but also advances significantly on [BS07], especially in Theorem 4.1.19 and Theorem 4.3.10. For a shorter treatment see also [Cur10].

4.1 General settings

Root bounds depend only on the length and the defining set. If we want bounds that improve on root bounds, we need to use other information on the code. In this section we introduce a new class of bounds, that we call `border bounds`, which use some knowledge on cyclic subcodes.

Definition 4.1.1. *A codeword c of a cyclic code C is called a* `border` *codeword for C if it is not contained in any proper cyclic sub-code of C.*
We denote by $\hat{C}$ the set of all border codewords of C.
We denote by $\hat{\mathrm{d}}(C)$ the `border distance` *of C, i.e. $\hat{\mathrm{d}}(C) = \min_{c\in\hat{C},c\neq 0} \mathrm{w}(c)$.*

The following lemma can be easily proved.

Lemma 4.1.2. *Let $C \in \mathcal{C}_{q,n}$. Let $c \in C$. Then c is a border codeword for exactly one cyclic sub-code D of C. The generator polynomial of D is the greatest common divisor of c and $x^n - 1 \in \mathbb{F}_q[x]$.*

The following fact is then obvious.

Fact 4.1.3. *For any $C \in \mathcal{C}$, we have*

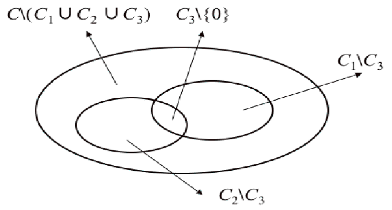
$$C = \cup_{D<C}\hat{D}, \quad \mathrm{d}(C) = \min_{D<C} \hat{\mathrm{d}}(D)\,.$$

Thanks to the previous fact, we can reformulate Theorem 2.1.8.

Proposition 4.1.4. *Let $C \in \mathcal{C}$. Let $\mathrm{DFT}(C)$ be the code formed by the Discrete Fourier Transforms of the words of C. Then the distance of C is*

$$\mathrm{d}(C) \;=\; \min_{D<C,D\neq\{0\}} \{ \min_{c\in\hat{D}} \mathrm{rk}(M(\mathrm{DFT}(c))) \}\,.$$

Figure 4.1: Border codewords



Definition 4.1.5. *We denote by $\mathcal{E}$ the subset of $\mathbb{N} \times 2^{\mathbb{N}} \times 2^{2^{\mathbb{N}}}$ s.t. $(n, S, \mathbf{S}) \in \mathcal{E}$ if $(n, S) \in \mathcal{D}$ and $\mathbf{S} = \{T_1, \ldots, T_s\}$, with $S \subsetneq T_h \subseteq \{0, \ldots, n-1\}$, $T_h \neq \{0, \ldots, n-1\}$ for any $1 \leq h \leq s$ and $T_h \neq T_k$ for $1 \leq h \neq k \leq s$.*

Let $(n, S, \mathbf{S}) \in \mathcal{E}$, $\mathbf{S} = \{T_1, \ldots, T_s\}$. We denote by $(n, S, \mathbf{S})^{\#}$ the set

$$(n, S, \mathbf{S})^{\#} \;=\; \{\{S_1, \mathbf{S}_1\}, \ldots, \{S_r, \mathbf{S}_r\}\},$$

where $r = |\mathbb{Z}_n^*|$, $\mathbf{S}_i = \{T_{i,1}, \ldots, T_{i,s}\}$ and for any $l \in \mathbb{Z}_n^*$ there is one and only one i such that $S_i = \{(lt)_n \mid t \in S\}$ and $T_{i,j} = \{(lt)_n \mid t \in T_j\}$ for any j.

Note that $\{S, \mathbf{S}\} \in (n, S, \mathbf{S})^{\#}$, $|S_h| = |S|$ for any h and $|T_{h,j}| = |T_j|$ for any j and any h. Using a function $\zeta \in \mathcal{Z}$, we define a map ψ_ζ from $\mathcal{C}$ to $\mathcal{E}$:

$$\psi_\zeta : \mathcal{C} \to \mathcal{E}, \quad \psi_\zeta(C) \;=\; (n, S_{C,\alpha}, \mathbf{S}), \mathbf{S} = \{S_{D,\alpha} \mid D < C,\; D \neq \{0\}\}, \tag{4.1}$$

where $\alpha = \zeta(\chi(C), n)$. In other words, $\psi_\zeta(C)$ contains the length of the code C, the defining set of C with respect to α and the defining sets of all the non-zero cyclic subcodes of C with respect to α. Note that we exclude the zero sub-code, which would give rise to the set $\{0, \ldots, n-1\}$, and that map ψ_ζ plays a role analogous to that of map ϕ_ζ in (3.1). We provide now a proposition, whose easy proof is omitted.

Proposition 4.1.6.

$$\forall\; \zeta, \zeta' \qquad \psi_\zeta(\mathcal{C}) = \psi_{\zeta'}(\mathcal{C}).$$

Thanks to the previous proposition, we define $\mathcal{E}' = \mathrm{Im}(\psi_\zeta)$ for any $\zeta \in \mathcal{Z}$. Note that $\mathcal{E}' \subseteq \mathcal{E}$.
Unlike ϕ_ζ, we have that for any ζ, ψ_ζ is not surjective, as show below.

Theorem 4.1.7.

$$\mathcal{E}' \subsetneq \mathcal{E}$$

Proof. Let $\zeta \in \mathcal{Z}$. We have to exhibit an $(n, S, \mathbf{S})$ such that $(n, S, \mathbf{S}) \in \mathcal{E}$ but $(n, S, \mathbf{S}) \notin \psi_\zeta(\mathcal{C})$. Let $n = 3$, $S = \{1\}$, $\mathbf{S} = \{\{1\}, \{1, 2\}\}$. If $(n, S, \mathbf{S}) \in \mathcal{E}'$, then $\{1\}$ is a cyclotomic coset and $\{1, 2\}$ is the union of cyclotomic cosets $\{1, 2\} = \{1\} \cup \{2\}$, but then also $\{3\}$ is a cosets and $\{1, 3\} \notin \mathbf{S}$. □

We are ready to define our new class of bounds.

Definition 4.1.8. *A* `border function` *is a map $f : \mathcal{E}' \to \mathbb{N} \cup \{\infty\}$ such that:*

$$\forall \zeta \in \mathcal{Z}, \forall\ C \in \mathcal{C}, \quad f \circ \psi_\zeta(C) \leq \mathrm{d}(C)\,. \tag{4.2}$$

We denote by $\mathcal{B}$ the class of all border functions.

Given $f \in \mathcal{B}$, we say that f is `invariant` *if $f(n, S, \mathbf{S}) = f(n, S', \mathbf{S}')$, for any $\{S', \mathbf{S}'\} \in (n, S, \mathbf{S})^\#$. We also denote by $f^\#$ the map*

$$f^\# = \max_{\{S', \mathbf{S}'\} \in (n,S,\mathbf{S})^\#} f(n, S', \mathbf{S}')\ .$$

For any $\zeta \in \mathcal{Z}$ and any $f \in \mathcal{B}$, the composite map $f_{\mathcal{E},\zeta} = f \circ \psi_\zeta : \mathcal{C} \mapsto \mathbb{N} \cup \{\infty\}$ is called the `border bound` *associated to f and ζ. If f is invariant, we say that $f_{\mathcal{E},\zeta}$ is invariant and we write $f_\mathcal{E}$. We denote by $\mathcal{B}_\mathcal{E}$ the class of all border bounds.*

Due to (4.2), border bounds are actually lower bounds for the distance on $\mathcal{C}$.

If $f \in \mathcal{B}$ is invariant, we have that $f_{\mathcal{E},\zeta} = f_{\mathcal{E},\zeta'}$ for any ζ and ζ' and so will just write $f_\mathcal{E}$. The following fact is then obvious.

Proposition 4.1.9. *For any $f \in \mathcal{B}$, $f^\#$ is invariant, $f \leq f^\#$ and $f^\#_\mathcal{E} = \max_{\zeta \in \mathcal{Z}} f_{\mathcal{E},\zeta}$.*

Proof. It is sufficient to adapt the arguments of Proposition 3.1.5. □

Remark 4.1.10. A root bound uses as information on the code only the length and the defining set. A border bound uses in addition the knowledge of defining sets of cyclic subcodes of C, which is the real meaning of parameter $\mathbf{S}$. It may not seem a significant gain but in practice border bounds outperforms root bounds.

If we take an arbitrary root function we can view it trivially as a border function, by ignoring parameter $\mathbf{S}$, as follows.

Definition 4.1.11. *Let $f \in \mathcal{R}$. We denote by $\bar{f}$ the map*

$$\bar{f} : \mathcal{E}' \longrightarrow \mathbb{N} \cup \{\infty\}, \qquad \bar{f} : (n, S, \mathbf{S}) \mapsto f(n, S)\,.$$

The following fact follows easily from (3.2) and (4.2).

Fact 4.1.12. *If $f \in \mathcal{R}$ then $\bar{f} \in \mathcal{B}$.*

We can thus view any root bound as a border bound, in principle, even if there is no point in computing one as such.

Remark 4.1.13. An invariant bound δ takes the same values on a code and any of its naturally equivalent codes, both when δ is a root bound and when δ is a border bound.

We can adapt many definitions and results for root functions to the "border case".

Definition 4.1.14. *Let f be a border function. We say that f is* `monotone` *if for any $(n, S, \mathbf{S})$ and $(n', S', \mathbf{S}')$ in $\mathcal{E}'$ we have*

$$n = n',\ S \subseteq S',\ \mathbf{S} \supseteq \mathbf{S}' \quad \Longrightarrow \quad f(n, S, \mathbf{S}) \leq f(n, S', \mathbf{S}')$$

Any border bound associated to f is called a `monotone` *border bound.*

Proposition 4.1.15. *Let f be any border function. We denote by f^* the map defined by $f^*(n, S, \mathbf{S}) = \max\{f(n, S', \mathbf{S}') \mid S' \subseteq S,\ \mathbf{S}' \supseteq \mathbf{S}\}$. We have: f^* is a border function, f^* is monotone, $f \leq f^*$. Moreover, if g is any monotone border functions s.t. $f \leq g$, then $f^* \leq g$.*

Proof. It is sufficient to adapt the argument of Proposition 3.1.11. □

It is now natural to introduce the `optimal border bound` and its characterization:

$$\mathsf{b}(n, S, \mathbf{S}) = \max\{f(n, S, \mathbf{S}) \mid f \in \mathcal{B}\}\,. \tag{4.3}$$

Theorem 4.1.16. *Map b is a border function, which is maximal in $\mathcal{B}$, monotone and invariant.*

Proof. See Theorem 3.1.12, using Proposition 4.1.15 and Proposition 4.1.9. □

Theorem 4.1.17. *Map $\mathsf{b}_{\mathcal{E}}$ is a monotone invariant root bound, which is maximal in $\mathcal{B}_{\mathcal{E}}$.*

Proof. It follows from Theorem 4.1.16. □

We have that, for any $\zeta \in \mathcal{Z}$ and any $(n, S, \mathbf{S}) \in \mathcal{E}'$:

$$\mathsf{b}(n, S, \mathbf{S}) = \min\{\mathrm{d}(C) \mid C \in \mathcal{C}, \psi_\zeta(C) = (n, S, \mathbf{S})\}.$$

The following theorem can be proved similarly to Theorem 3.1.18

Theorem 4.1.18. *For any $\zeta \in \mathcal{Z}$, we have*

$$\begin{aligned} \mathsf{b}_{\mathcal{E},\zeta}(C) = & \min\{\mathrm{d}(C') \mid C' \in \mathcal{C}, \psi_\zeta(C') = \psi_\zeta(C)\}\,, \\ \mathsf{b}_{\mathcal{E},\zeta}(C) = & \mathsf{b}_{\mathcal{E}}(C) = \max_{\zeta' \in \mathcal{Z}} \mathsf{b}_{\mathcal{E},\zeta'}(C) = \end{aligned}$$

$$\max\nolimits_{1 \le i \le r}\{\min\{\mathrm{d}(C') \mid C' \in \mathcal{C}, S_{C',\beta} = S_{C,\alpha_i}, \mathbf{S_{C',\beta}} = \mathbf{S_{C,\alpha_i}}, \alpha_i = \zeta(\chi(C),n), \beta = \zeta(\chi(C'),n)\}\}\,,$$

where $C \in \mathcal{C}_{q,n}$ and $\alpha_1, \ldots, \alpha_r$ are all primitive n-th roots of unity over $\mathbb{F}_q$.

We are also able to prove the analogous of Theorem 3.1.20

Theorem 4.1.19.

$$\mathsf{b}_{\mathcal{E}} \neq \mathrm{d}\,.$$

Proof. For the proof, we follow the strategy used in the proof of Theorem 3.1.20: first, we suppose the existence of two codes with some properties which are enough to prove our claim, second we provide explicitly such codes.

Part I

We look for two codes C_1, C_2 of length $n \ge 1$, over $\mathbb{F}_{q_1}$ and over $\mathbb{F}_{q_2}$ respectively, s.t.

- the two fields have different characteristics, let us say $\chi(C_1) = p_1$ and $\chi(C_2) = p_2$, with $p_1 \neq p_2$,
- $\alpha_1, \ldots, \alpha_r$ are all the primitive $n-$th roots of unity over $\mathbb{F}_{q_1}$ and $\beta_1, \ldots, \beta_r$ are all the primitive $n-$th roots of unity over $\mathbb{F}_{q_2}$.

We take $\zeta_1, \ldots, \zeta_r \in \mathcal{Z}$ s.t. $\zeta_i(\chi(C_1), n) = \alpha_i$ and $\zeta_i(\chi(C_2), n) = \beta_i$, for $1 \le i \le r$. We also want:

$$\mathrm{d}(C_1) < \mathrm{d}(C_2) \tag{4.4}$$

$$S_{C_1,\alpha_i} = S_{C_2,\beta_i}, \quad 1 \le i \le r \tag{4.5}$$

$$\mathbf{S_{C_1,\alpha_i}} = \mathbf{S_{C_2,\beta_i}}, \quad 1 \le i \le r. \tag{4.6}$$

With the above assumptions on C_1 and C_2, we now prove the first part. Note that (4.5) and (4.6), implies $\psi_{\zeta_i}(C_1) = \psi_{\zeta_i}(C_2)$, for any $1 \le i \le r$. From Theorem 4.1.18 we have

$$\mathsf{b}_{\mathcal{E}}(C_2) = \max_{1 \le i \le r} \{\, \min\{\, \mathrm{d}(C) \mid C \in \mathcal{C},\ \psi_{\zeta_i}(C) = (n, S_{C_1,\alpha_i}, \mathbf{S_{C_1,\alpha_i}}\,\}\,\}\,. \tag{4.7}$$

However $C_1 \in \mathcal{C}$ and $\psi_{\zeta_i}(C_1) = (n, S_{C_1,\alpha_i}, \mathbf{S_{C_1,\alpha_i}})$ for any $1 \le i \le r$, so

$$\mathsf{b}_{\mathcal{E}}(C_2) \le \max_{1 \le i \le r} \mathrm{d}(C_1) = \mathrm{d}(C_1) < \mathrm{d}(C_2). \tag{4.8}$$

Part II

We take $\mathbb{F}_{q_1} = \mathbb{F}_{3^5}$ and $\mathbb{F}_{q_2} = \mathbb{F}_{2^{10}}$, $n = 11$. There are $r = 10$ primitive $n-$th roots of unity. As cyclic codes C_1 and C_2 we take two codes with defining set $S = S_{C_1,\alpha_1} = S_{C_2,\beta_1}$,

$$S = \{\, 0, 1, 2, 3, 5 \,\} .$$

Since $\mathbb{F}_{2^{10}}$ and $\mathbb{F}_{3^5}$ are the splitting field of $x^{11}-1$ over $\mathbb{F}_2[x]$ and $\mathbb{F}_3[x]$, respectively,we have that the subcodes are

$$\mathbf{S}_{\mathbf{C_1},\alpha_\mathbf{i}} = \{\, S \subseteq T \subseteq \{0, \ldots, 10\} \mid T \neq \{0, \ldots, 10\} \,\} = \mathbf{S}_{\mathbf{C_2},\beta_\mathbf{i}}$$

for any $1 \leq i \leq r$. An explicit computation (see Section 9.3) shows that $\mathrm{d}(C_1) = 5$ and $\mathrm{d}(C_2) = 6$. □

Theorem 4.1.19 is a generalization of Theorem 3.1.20 , since we can see any root function as a border function. We think that this result is relevant. In fact, since border bounds use also information on the subcodes to estimate the distance, someone may think that, differently from the root bounds, they are able to reach the true distance for a code. Unfortunately, Theorem 4.1.19 states that if you get a bound which uses only the information given by the defining set of the codes and all the defining sets of the (non-trivial) subcodes, there is at least a code for which you cannot reach the true distance. In other word, you cannot get the distance of a code without knowing the characteristic of the field. The questions proposed in Section 3.1 about the optimal root bound can be studied in the more general framework of the optimal border bound.

4.2 Border bounds and $\mathcal{U}$

The advantage of using border functions instead of root functions is that all border codewords of a code share in their DFT not only the same zeros, but also the same non-zeros. So the rank evaluation in $\mathcal{U}$ can be more precise, since the involved matrices have entries only in $\{\Delta^{+}, 0\}$. In this section we provide several methods for bounding the rank of this type of matrices.

We formalize how it is possible to perform the rank evaluation in $\mathcal{U}$ introducing the concept of localization map.

Definition 4.2.1. *Let $\hat{f} : \mathcal{D} \longrightarrow \mathbb{N} \cup \{\infty\}$ be any function s.t.*

$$\hat{f}(n, T) \leq \mathrm{rk}(M(\hat{R}(n, T))) .$$

Then $\hat{f}$ is called `localization`.

So a localization map is a way to bound the rank of matrices over $\mathcal{U}$, exploiting the fact that they have entries only in $\{\Delta^{+}, 0\}$.

Localizations present in literature are constructed by employing an *independence-check procedure*, as follows.

Definition 4.2.2. *Let ϵ be an algorithm that admits as input any matrix $A \in \mathcal{U}^{n\times m}$ over $\mathcal{U}$ and that returns either* true *or* false. *We say that ϵ is an* ***`independence-check procedure`*** *if any time it returns* true *then its input A has maximal rank, i.e.* $\mathrm{rk}(A) = \min(n, m)$.

Remark 4.2.3. When an independence-check procedure returns false, the matrix rows might be independent. In designing an independence-check procedure a trade-off has to be sought between time-consuming checks, that return true on a large number of independent sets, and fast checks, that may not recognize many independent sets but allow efficient implementations.

Given an independence-check procedure ϵ, we can construct algorithms to get a lower-estimate on the rank of an arbitrary matrix over $\mathcal{U}$, which we call **`rank-bounding algorithms`**, by checking the rank of specific row subsets.
In literature two rank-bounding algorithms can be found (implicitly or explicitly), which we call the `first` and `second rank-bounding` algorithm[1].

Remark 4.2.4. Different rank-bounding algorithms from independence-check procedure scan different row subsets. Although one would expect to get different values in most cases, this is not so obvious when a matrix is of the type $A = M(\mathbf{v})$. For example, for most binary cyclic codes with $n \leq 63$ the bounds obtained from the first and second algorithm output the same value (and we do not know their output on the others). However, we have found an explicit example (see Section 9.3) in which the first and second algorithm output different values, on a matrix of kind $M(\mathbf{v})$, as follows.
Let $\mathbf{v} = (0, 0, 0, \Delta^{+}, 0, \Delta^{+}, 0, 0, \Delta^{+}, \Delta^{+}, \Delta^{+}) \in \mathcal{U}^{11}$. The first rank-bounding algorithm applied to $M(\mathbf{v})$ returns 5, the second returns 6 (with the set of rows $\{1, 2, 3, 4, 7, 8\}$).

First rank-bounding algorithm from ϵ

`Input`

A matrix A over $\mathcal{U}$.

`First Step`

We initialize a list S of rows of A with the first row.

`Cycle`

We call ϵ on the submatrix formed by S.

- If it returns *true*, then the rows of S are linearly independent,
- else we discard the last row of S.

[1] recently, also a third rank bounding algorithm has been presented in [ZK10, ZK11]

If there are other rows in A, we add to S the first row of A that has not been considered yet and we cycle again. Else we return the number of rows of S.

Second rank-bounding algorithm from ϵ

`Input`

A matrix A over $\mathcal{U}$.

`First Step`

We initialize $\ell = 1$.

`Cycle`

We call ϵ on all subsets of ℓ rows (with some order).

- If ϵ returns *true* on any subset, then we set $\ell = \ell + 1$ and restart the cycle, unless $\ell = n$, in which case we return n.
- else we return $\ell - 1$.

4.3 Strict border bounds

Once it is clear how to work with localizations, many definitions and results for strict root functions and strict root bounds can be adapted to the "border case". Since many of the results we cite are already proved in Section 3.3, we just mention them, providing a proof only for a few.

Using the localization maps we are now able to define the analogous of strict root functions and strict root bounds, in the border framework.

Definition 4.3.1. *A border function is called a* `strict border function` *if there is a localization $\hat{f} : \mathcal{D} \longrightarrow \mathbb{N} \cup \{\infty\}$ s.t.*

$$f(n, S, \mathbf{S}) = \min_{T \in \mathbf{S}} \hat{f}(n, T), \qquad \hat{f}(n, T) \leq \mathrm{rk}(M(\hat{R}(n, T))) \ .$$

In this case, we say that $\hat{f}$ is called the `localization` *of f.*

Any border bound associated to a strict border function is called a `strict border bound`.

We denote by $\mathcal{B}^S$ the class of all strict border functions and with $\mathcal{B}^S_{\mathcal{E}}$ the class of all strict border bounds.

Remark 4.3.2. A strict border bound computes a rank bound for all cyclic subcodes and then it takes the minimum. Moreover, the same function (the localization) is used for all cyclic subcodes. This further requirement could be relaxed, but with care, because of the exponential growth of the number of subcodes.

With an adaption of the argument in Proposition 3.3.2, it is easy to prove:

Proposition 4.3.3. *Any strict border function is a border function. That is*

$$\mathcal{B}^S \subseteq \mathcal{B}.$$

It is easy to see that a strict root function is actually a strict border function (with the $\bar{f}$ construction).

What differentiates a strict border bound from another is the localization map, i.e. the way they bound the rank of matrices over $\mathcal{U}$ (with entries only in $\{0, \Delta^{\!+}\}$). In the following definition we formalize the connection between strict root functions, localizations, rank-bounding algorithms and independence-check procedures.

Definition 4.3.4. *Let f be a strict root function and $\hat{f}$ be its localization. Let ϵ be an independence-check procedure, $\dot{\epsilon}$ be its first rank-bounding algorithm, and $\tilde{\epsilon}$ be its second rank-bounding algorithm.*
We say that f (or $\hat{f}$) is the `first realization` of ϵ, if the value $\hat{f}(n, T)$ is computed by applying $\dot{\epsilon}$ to $M(\hat{R}(n, T))$ for any $(n, T) \in \mathcal{D}$.
The `second realization` is defined analogously from $\tilde{\epsilon}$.
In both cases, we say that f is `based` on ϵ.

Given an independence-check procedure ϵ, Definition 4.3.4 explains how to compute a strict border bound: we obtain the defining sets of all cyclic subcodes, we construct for any of these a matrix $M(\hat{R}(n, S))$, we calculate a bound on $\mathrm{rk}(M(\hat{R}(n, S))$ via successive applications of ϵ to subsets of rows of $M(\hat{R}(n, S))$, and finally we take the minimum of these values.

Remark 4.3.5. All root bounds explicitly presented in Chapter 3 are polynomial-time bounds (in the length), except (possibly) for the optimal bound.
On the contrary, any border bound based on an independence-check procedure is at least time-exponential, since it has to examine all cyclic subcodes.
So, generally speaking, bounds like the Schaub bound and the VW shifting bound (see Section 4.4) outeperform classical bounds (i.e. root bounds), but at the price of larger input information and of drastically longer computations.

We can define the `invariant strict border functions (bounds)`.

Proposition 4.3.6. *For any strict border function, f, we denote by $f^{\#}$ the map defined by $f^{\#}(n, S, \mathbf{S}) = \max_{\{T, \mathbf{T}\} \in (n, S, \mathbf{S})^{\#}} f(n, T, \mathbf{T})$. Then: $f^{\#} \in \mathcal{B}^S$, $f^{\#}$ is invariant, $f \leq f^{\#}$, $f^{\#}_{\mathcal{E}} = \max_{\zeta \in \mathcal{Z}} f_{\mathcal{E}, \zeta}$.*

Proof. Adapt the arguments of Proposition 3.3.3. □

We can define `monotone strict border functions (bounds)`.

Proposition 4.3.7. *Let f be any strict border function. We denote by f^* the map defined by $f^*(n,S,\mathbf{S}) = \max\{f(n,S',\mathbf{S}') \mid S' \subseteq S,\ \mathbf{S}' \supseteq \mathbf{S}\}$. We have that f^* is a strict border function, f^* is monotone, $f \leq f^*$. Moreover, if g is any monotone strict border function s.t. $f \leq g$, then $f^* \leq g$.*

Proof. Adapt the arguments of Proposition 3.3.4. □

We can define the `maximal strict border function (bound)`.

$$\mathsf{b}^S(n,S,\mathbf{S}) = \max\{f(n,S,\mathbf{S}) \mid f \in \mathcal{B}^S\}\,. \tag{4.9}$$

Theorem 4.3.8. *Map b^S is a strict border function, which is maximal in $\mathcal{B}^S$, monotone and invariant.*

Proof. Adapt the arguments of Theorem 3.3.6. □

We can characterize the maximal strict border function (bound).

Theorem 4.3.9.
$$\mathsf{b}^S(n,S,\mathbf{S}) = \min_{T\in\mathbf{S}}\{\mathrm{rk}(M(\hat{R}(n,T)))\}.$$

Proof. Let $c(n,S,\mathbf{S}) = \min_{T\in\mathbf{S}}\{\mathrm{rk}(M(\hat{R}(n,T)))\}$. Clearly, c is a strict border function and its localization is $\hat{c}(n,T) = \mathrm{rk}(M(\hat{R}(n,T)))$. Our claim follows immediately by noting that $\mathrm{rk}(M(\hat{R}(n,T))) = \hat{c}(n,T) \leq \hat{\mathsf{b}}^S(n,T)$ since b^S is maximal, and $\hat{\mathsf{b}}^S(n,T) \leq \mathrm{rk}(M(\hat{R}(n,T)))$, because $\mathsf{b}^S \in \mathcal{B}^S$. □

Finally, we are able to prove the analogous of Theorem 3.5.8, still using the result of Theorem 3.5.7.

Theorem 4.3.10.
$$\mathsf{b}^S < \mathsf{b}$$

Proof. It is clear that $\mathsf{b} \geq \mathsf{b}^S$. We have only to exhibit an $(n,S,\mathbf{S}) \in \mathcal{E}'$ such that $\mathsf{b}^S(n,S,\mathbf{S}) < \mathsf{b}(n,S,\mathbf{S})$. Let us consider the map f_{Roos} of Definition 3.5.2. We have $f_{\mathrm{Roos}} \in \mathcal{R}$ and then $\bar{f}_{\mathrm{Roos}} \in \mathcal{B}$ so, $\bar{f}_{\mathrm{Roos}} \leq \mathsf{b}$. Let C be the code in $\mathcal{C}_{q^m,20}$, where $\mathbb{F} = \mathbb{F}_{q^m}$ is the splitting field of $x^{20}-1$ over $\mathbb{F}_q[x]$, with $S = \{0,1,6,7,9,10\}$. Let $\zeta \in \mathcal{Z}$ be such that $\psi_\zeta(C) = (20,S,\mathbf{S})$, where $\mathbf{S} = \{\, T \mid T \subsetneq \{0,\ldots,19\},\ S \subseteq T \,\}$. From Theorem 4.3.9 and Theorem 3.5.7 we have

$$\mathsf{b}^S(n,S) = \min_{T\in\mathbf{S}}\{\mathrm{rk}(M(\hat{R}(n,T)))\} \leq 4$$

while $\mathsf{b}(n, S, \mathbf{S}) \geq \bar{f}_{\mathrm{Roos}}(n, S, \mathbf{S}) = 5$, hence

$$\mathsf{b}^S(n, S, \mathbf{S}) \leq 4 < 5 = \bar{f}_{\mathrm{Roos}}(n, S, \mathbf{S}) \leq \mathsf{b}(n, S, \mathbf{S}).$$

A similar proof holds also if we consider the map f_{B5} of Definition 3.5.5 with defining set $S' = \{0, 1, 3, 4, 6, 7\}$ and the code $C' \in \mathcal{C}_{q^m,13}$, where $\mathbb{F}_{q^m} = \mathbb{F}$ is the splitting field of $x^{13} - 1$ over $\mathbb{F}_q[x]$. □

Remark 4.3.11. Note that an obvious consequence of Theorem 4.3.10 is that $\mathcal{B}^S \subsetneq \mathcal{B}$. In particular, considering f_{Roos} and f_{B5}, we have $\bar{f}_{\mathrm{Roos}}, \bar{f}_{\mathrm{B5}} \in \mathcal{B}$, but $\bar{f}_{\mathrm{Roos}}, \bar{f}_{\mathrm{B5}} \notin \mathcal{B}^S$.

We conclude this section observing that there are some bounds which are based on the `unknown syndromes` (see e.g. [FT91a], [FT91b], [FT89], [MAI97]). Clearly, any syndrome matrix can be transformed into a matrix over $\mathcal{U}$, where a Δ is inserted to replace an unknown syndrome and a $\not\Delta$ is inserted to replace a known syndrome. These bounds can then be translated as operating on this matrix and therefore we think that they may be either strict root bounds or strict border bounds. However, we do not provide here a detailed description.

4.4 Equivalence of border bounds

Here we show that all known localizations are based on the same ϵ, which turns out to be equivalent tothe singleton procedure introduced in Section 3.2. In the remainder of the section we describe the singleton-procedure bound and the Schaub bound, proving that they are equivalent. Finally, we describe the famous Van Lint-Wilson shifting bound ("VW bound"), proposed in [vLW86], and we show that it is closely related to the singleton procedure bound (and hence to the Schaub bound due to Theorem 4.4.4).

We would also like to mention an alternative but unpublished independence-check procedure, due to F. Ponchio and M. Sala ([PS03]) which uses more deeply the underlying field structure.

We consider the singleton procedure described in Section 3.2. By Theorem 3.2.6, this is obviously an independence-check procedure, as formalized in Definition 4.2.2. From now on, we indicate the singleton procedure as ϵ_s. We have that ϵ_s plays a special role, with respect to all other possible independence-check procedures, in fact we can easily show that any result of independence for vectors in $\mathcal{U}$ can be obtained using the singleton procedure.

Proposition 4.4.1. *Let ϵ, ϵ_s be any independence check procedure and the singleton independence-check procedure respectively. If ϵ returns true then also ϵ_s returns true.*

Proof. Let $A \in \mathcal{U}^{n\times m}$ be an input matrix for ϵ. If ϵ returns true then its input A has maximal rank. Let $t = \min(n, m)$ be the rank of A, then for Theorem 3.2.18 $\mathrm{rk}(A) = \mathrm{prk}(A)$, which means that ϵ_s is successful for A. □

4.4.1 The Schaub bound

This bound was first presented in [Sch88].

First, we describe the independence-check procedure proposed by Schaub.

The Schaub independence-check procedure

`Input`

A matrix A over $\mathcal{U}$, whose rows are n-dimensional vectors in $\mathcal{U}^n$ and form a set $R = \{r_1, \dots, r_h\}$. We can assume that all vectors except r_h form a linearly independent set.

`Initialization`

We consider $h - 1$ unknowns values in $\mathcal{U}$: $\{c_1, \dots c_{h-1}\}$.

`Cycle`

For any column i of A, we must have

$$r_h(i) = \sum_{j=1}^{h-1} c_j \cdot r_j(i)\,,$$

for some c_j (not depending on i). We deduce from this relation the values in $\mathcal{U}$ that the c_j can have, using also the relevant information obtained from the previous columns $\{1, \dots, i - 1\}$.

If we find some contradiction for at least one of the c_j, then we are sure that the rows are linearly independent and so we return *true*.

Otherwise we pass to the next column.

`Last step`

We return *false*, because no contradiction arose.

To understand how it works, we propose the following example, where the associated first rank-bounding algorithm is applied. This same example is redone with the "singleton" independence-check procedure (Example 4.4.3), so that the efficiency improvement given by the latter is apparent.

Example 4.4.2. Let T be the matrix (over $\mathcal{U}$)

$$T \quad = \quad \begin{pmatrix} \Delta^{+} & \Delta^{+} & 0 & 0 & \Delta^{+} & 0 & \Delta^{+} & \Delta^{+} \\ 0 & 0 & 0 & 0 & 0 & \Delta^{+} & 0 & 0 \\ 0 & 0 & \Delta^{+} & \Delta^{+} & 0 & 0 & \Delta^{+} & 0 \\ \Delta^{+} & \Delta^{+} & 0 & 0 & 0 & \Delta^{+} & 0 & 0 \\ 0 & 0 & 0 & \Delta^{+} & \Delta^{+} & 0 & 0 & \Delta^{+} \\ 0 & 0 & \Delta^{+} & \Delta^{+} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta^{+} & 0 & 0 & 0 \end{pmatrix}$$

We need to apply several times the Schaub independence-check procedure to estimate $\mathrm{rk}(T)$. We shorten "the Schaub independence-check procedure" to "the procedure" in this example. We name the rows consecutively $\{r_1, \dots, r_7\}$.

- It is obvious that $\mathrm{rk}(T) \geq 1$, since the first row contains some Δ^{+}'s.

- We apply the procedure to the first two rows: we try to see the second row as a linear combination of the first one. Let $c_1 \in \mathcal{U}$ s.t. $r_2 = c_1 \cdot r_1$. The first column gives $r_2(1) = c_1 \cdot r_1(1)$, i.e. $0 = c_1 \cdot \Delta^{+}$ and hence $c_1 = 0$. The sixth column gives $r_2(7) = c_1 \cdot r_1(7)$, that is, $\Delta^{+} = 0 \cdot 0 = 0$.
  This is clearly impossible, so the rows are independent and so $\mathrm{rk}(T) \geq 2$.

- We try to see r_3 as a linear combination of $\{r_1, r_2\}$. We impose $r_3 = c_1 \cdot r_1 + c_2 \cdot r_2$. The first column gives $r_3(1) = c_1 \cdot r_1(1) + c_2 \cdot r_2(1)$, i.e. $0 = c_1 \cdot \Delta^{+} + c_2 \cdot 0 = c_1 \cdot \Delta^{+}$, which restricts $c_1 = 0$.
  The second column is equal to the first, so it can give no more information and we skip it. From now we will skip equal columns without any further comment. The third column gives $r_3(3) = c_1 \cdot r_1(3) + c_2 \cdot r_2(3)$, i.e.
  $\Delta^{+} = 0 \cdot 0 + c_2 \cdot 0 = 0$. This is impossible, so $\mathrm{rk}(T) \geq 3$.

- We write $r_4 = c_1 \cdot r_1 + c_2 \cdot r_2 + c_3 \cdot r_3$. The first column gives $\Delta^{+} = c_1 \cdot \Delta^{+}$, i.e. $c_1 = \Delta^{+}$. Third column: $0 = c_3 \cdot \Delta^{+}$, i.e. $c_3 = 0$.
  Fifth column: $\Delta^{+} \cdot \Delta^{+} = 0$, i.e. $\Delta^{+} = 0$, impossible and so $\mathrm{rk}(T) \geq 4$.

- With similar computations we can prove the linear independence of the first five rows and hence $\mathrm{rk}(T) \geq 5$.

- We write $r_6 = \sum_{j=1}^{5} c_i \cdot r_i$. The first column gives

$$0 = c_1 \cdot \Delta^{+} + c_4 \cdot \Delta^{+} \tag{4.10}$$

  We take note of this constraint and proceed.
  Third column: $\Delta^{+} = c_3 \cdot \Delta^{+}$, i.e. $c_3 = \Delta^{+}$.

Fourth column: $\Delta^{\!\!\!+} = \Delta^{\!\!\!+} \cdot \Delta^{\!\!\!+} + c_5 \cdot \Delta^{\!\!\!+}$, i.e $\Delta^{\!\!\!+} = \Delta^{\!\!\!+} + c_5 \cdot \Delta^{\!\!\!+}$, i.e $c_5 = \Delta$ (no information on c_5).
The fifth and sixth columns, respectively, give:

$$0 = c_1 \cdot \Delta^{\!\!\!+} + c_5 \cdot \Delta^{\!\!\!+} \tag{4.11}$$

$$0 = c_2 \cdot \Delta^{\!\!\!+} + c_4 \cdot \Delta^{\!\!\!+} \tag{4.12}$$

Seventh column: $0 = c_1 \cdot \Delta^{\!\!\!+} + \Delta^{\!\!\!+} \cdot \Delta^{\!\!\!+}$, i.e. $c_1 = \Delta^{\!\!\!+}$.
Then we have by (4.10) $c_4 = \Delta^{\!\!\!+}$ and by (4.11) $c_5 = \Delta^{\!\!\!+}$. But then by (4.12), we have $c_2 = \Delta^{\!\!\!+}$. Eighth column: $0 = \Delta^{\!\!\!+} \cdot \Delta^{\!\!\!+} + \Delta^{\!\!\!+} \cdot \Delta^{\!\!\!+}$, no contradiction.
No contradiction arises: we discard the sixth row.

- We write $r_7 = \sum_{j=1}^{5} c_i \cdot r_i$. First column:

$$0 = c_1 \cdot \Delta^{\!\!\!+} + c_4 \cdot \Delta^{\!\!\!+} \tag{4.13}$$

  Third column: $0 = c_3 \cdot \Delta^{\!\!\!+}$, i.e. $c_3 = 0$.
  Fourth column: $0 = 0 \cdot \Delta^{\!\!\!+} + c_5 \cdot \Delta^{\!\!\!+}$, i.e. $c_5 = 0$.
  Fifth column: $\Delta^{\!\!\!+} = c_1 \cdot \Delta^{\!\!\!+}$, i.e. $c_1 = \Delta^{\!\!\!+}$ and we get $c_4 = \Delta^{\!\!\!+}$ by (4.13).
  Sixth column: $0 = c_2 \cdot \Delta^{\!\!\!+} + \Delta^{\!\!\!+}$, i.e. $c_2 = \Delta^{\!\!\!+}$.
  Seventh column: $0 = \Delta^{\!\!\!+} \cdot \Delta^{\!\!\!+} + 0 \cdot \Delta^{\!\!\!+} = \Delta^{\!\!\!+}$, impossible, so $\mathrm{rk}(T) \geq 6$.

The seventh row was the last, the final result is $\mathrm{rk}(T) \geq 6$.

From the Schaub independence-check procedure one can directly obtain its first realization (Definition 4.3.4) and view the latter as the localization of a strict border function, which we call the "Schaub function" for short. It is obvious how to do that and so we do not detail it. We conclude that the so-called "Schaub bound" ([Sch88]) is nothing else but the border bound associated to the Schaub function.

4.4.2 The singleton-procedure bound

This bound has been presented in [Sal01], but the notation there is quite different and not easy to follow.

Let us consider the first rank-bounding algorithm of the singleton procedure, $\dot{\epsilon}_s$. We define a strict border function h as the first realization of ϵ_s. We say that h is the `singleton-procedure function`. The strict border bound associated to h is the singleton-procedure bound.

To visualize how it works, it is instructive to examine the following example, which is Example 4.4.2 redone using $\mathbf{h}$.

Example 4.4.3. We consider the matrix T present in Example 4.4.2.

- It is obvious that $\mathrm{rk}(T) \geq 1$.
- Rows $\{r_1, r_2\}$. The first column is a singleton. Removing it produces a non-zero row, i.e. $\mathrm{rk}(T) \geq 2$
- Rows $\{r_1, r_2, r_3\}$. In order, singletons found (and removed): col. 1 and 3, so $\mathrm{rk}(T) \geq 3$.
- Rows $\{r_1, r_2, r_3, r_4\}$. In order, singletons found: col. 3, col. 5 and col. 1, hence $\mathrm{rk}(T) \geq 4$.
- Rows $\{r_1, r_2, r_3, r_4, r_5\}$. Singletons: $\{3, 4, 5, 7\}$, so $\mathrm{rk}(T) \geq 5$.
- Rows $\{r_1, r_2, r_3, r_4, r_5, r_6\}$. There are no singletons: we discard the row, so that we still have $\mathrm{rk}(T) \geq 5$.
- Rows $\{r_1, r_2, r_3, r_4, r_5, r_7\}$. Singletons: $\{3, 7, 8, 5, 1\}$, thus $\mathrm{rk}(T) \geq 6$.

4.4.3 Singleton-procedure bound and Schaub bound are equivalent

This subsection is devoted to showing that the Schaub bound is equivalent to the singleton-procedure bound.

Theorem 4.4.4. *The localization maps of the Schaub function and of the singleton-procedure function are based on the same independence-check procedure, of which they are the first realization. As a consequence, for any choice of $\zeta \in \mathcal{Z}$ their associated border bounds are equivalent.*

Proof. In the following we shorten "the singleton independence-check procedure" to "the singleton procedure" and "the Schaub independence-check procedure" to "the Schaub procedure".

Let S be an $h \times n$ matrix over $\mathcal{U}$, $1 \leq h \leq n$. We denote by:

- $\mathcal{M}_S$, the following logical statement $\{$*we can prove that the rows of S are a linearly independent set by applying Schaub's procedure*$\}$,
- $\mathcal{Q}_S$, the following logical statement $\{$*we can prove that the rows of S are a linearly independent set by applying the singleton procedure*$\}$.

We will also denote by $\{s_1, \ldots, s_h\}$ the h rows of S, where $s_j = (s_j(1), \ldots, s_j(n))$. Note that the statement of Theorem 4.4.4 can be rephrased as $\mathcal{Q}_S \iff \mathcal{M}_S$.

$\mathcal{Q}_S \implies \mathcal{M}_S$ **.** By induction on the number of rows, h. For $h = 1$ it is obvious. By inductive hypothesis, we suppose that the implication is true for $h - 1$ rows. We prove it holds for h rows. Let $1 \leq j \leq n$ be a column index such that the $j-$th column is a singleton, with corresponding row s_i, with $s_i \in \{ s_1, \ldots, s_h \}$, i depending on j. We have $s_i(j) = \Delta\!\!\!^{+}$ and $s_k(j) = 0$ for $k \neq i$. Then for any $c_1, \ldots, c_{h-1} \in \mathcal{U}$ we have:

$$\Delta\!\!\!^{+} = s_i(j) = \sum_{k \neq j} c_k s_k(j) = 0,$$

which is impossible. Thus s_i is linearly independent from the other rows and by inductive hypothesis we conclude.

$\mathcal{M}_S \implies \mathcal{Q}_S$ **.** It follows immediately from Proposition 4.4.1, taking as ϵ the Schaub procedure.

□

4.4.4 On the Van-Lint Wilson shifting bound

To describe the VW bound we need the following definition.

Definition 4.4.5 ([vLW86])**.** *Let $n \geq 1$ be an integer number. Let S be a subset of $\{0, \ldots, n-1\}$. We say that A is independent from S if:*

1. *A is the empty set,*

2. *A is a shift of an independent set B, i.e. if B is independent with respect to S and $c \in \{0, \ldots n-1\}$, then $A = c + B = \{(c+b)_n \mid b \in B\}$ is independent.*

3. *A is $B \sqcup \{a\}$, with B independent and included in S, and $a \notin S$.*

Example 4.4.6. Let $S = \{1, 2, 4\} \subseteq \{0, \ldots, 6\}$.

- By 1, we have that $A^{(0)} = \emptyset$ is an independent set.

- By 3, we have that $A^{(1)} = \{3\}$ is also an independent set, since $A^{(1)} = \{3\} \cup A^{(0)}$, with $3 \notin S$, $A^{(0)}$ independent.

- By 2, $A^2 = \{1\}$ is independent, because $A^{(2)} = 4 + A^{(1)}$.

- $A^{(3)} = \{1, 2\}$ is independent, in fact $A^{(3)} = 1 + (A^{(2)} \sqcup \{6\})$.

- $A^{(4)} = \{1, 2, 3\}$ is independent, because $A^{(4)} = A^{(3)} \sqcup \{3\}$.

By an exhaustive search we find no independent sets with size greater than 3.

The VW bound can be described algorithmically as follows.

```
Van Lint-Wilson shifting bound
Input
A cyclic code $C \in \mathcal{C}_{q,n}$ and $\alpha$, where $\alpha$ is a primitive n-th root of unity over $\mathbb{F}_q$.

Cycle
For any cyclic subcode $D$ of $C$.
Compute $S = S_{D,\alpha}$.
Compute the length $\lambda(D)$ of the largest set independent from $S$.

Last step
Output $\min_{D<C} \lambda(D)$.
```

Remark 4.4.7. This bound is not formalized in [vLW86], where one can find a theorem linking distance and length of independent sets, with a few examples that are supposed to illuminate the use of the theorem. In particular, the fact that all cyclic subcodes of the code have to be considered is not immediately apparent, since the examples present lucky cases where only a few subcodes are needed.

From our description it is clear that the VW bound is a border bound, requiring a computation for any cyclic subcode. We claim much more, i.e. that it is a strict border bound and that it is strongly linked to the other two known border bounds. To be more precise, we claim the following.

Theorem 4.4.8. *The VW bound is a strict border bound. The localization of its strict border function coincides with the second realization of the singleton procedure.*

We recall that ϵ_S indicates the singleton-procedure. Since obviously the localization of the VW function is the size of the largest set independent from the defining set, to prove Theorem 4.4.8 it is sufficient to show the following proposition.

Proposition 4.4.9. *Let $C \in \mathcal{C}$ and $\zeta \in \mathcal{Z}$. Let $S_{C,\alpha}$, with $\alpha = \zeta(\chi(C), n)$. Let λ be the size of the largest set independent from $S_{C,\alpha}$. Let r be the output $\tilde{\epsilon}_s$ (the second realization of ϵ_s) applied to $M = M(\hat{R}(n, S_{C,\alpha}))$. Then*

$$r = \lambda\,.$$

Before proving Proposition 4.4.9, we give a couple of lemmas.

Lemma 4.4.10. *Let A, S be non-empty sets. If A is independent of S, then there is another set B and an element $a \notin S$ such that A is a shift of $B \sqcup \{a\}$ and B is independent from S but $B \subseteq S$.*

Proof. It is a direct consequence of Definition 4.4.5 (since A is non-empty). □

Lemma 4.4.11. *Let $S \subseteq \{0, \dots, n-1\}$, $S \neq \emptyset$. Let $\mathbf{w} = \hat{R}(n, S))$ and $M = M(\mathbf{w})$. Let $\mathbf{v} = M[1]$ be the first column of M. Then for any $1 \leq i \leq n$ we have $\mathbf{v}[i] = \mathbf{w}[(n-i+1)_n + 1]$.*

Proof. It follows immediately from the circularity of M. □

To ease our notation in the remainder of this sub-section, when we deal with any integer b we implicitly mean $(b)_n$, so that previous lemma may be stated as $\mathbf{v}[i] = \mathbf{w}[n-i+2]$.

Lemma 4.4.12. *Let $T, S \subseteq \{0, \dots, n-1\}$ be non-empty sets, with $T = \{t_1, \dots, t_\tau\}$, $S = \{s_1, \dots, s_h\}$. Let $a \notin S$. Let $M = M(\hat{R}(n, S))$ be formed by rows $M_1, \dots, M_n$. Let $\bar{M} = \{M_{n-a+2}, M_{n-t_1+2}, \dots, M_{n-t_\tau+2}\}$ be a sub-matrix of M. Then*

$$\bar{M}[1] \quad \text{is a singleton} \qquad \Longleftrightarrow \qquad \mathrm{T} \subseteq \mathrm{S}\,.$$

Proof. Let $\bar{M} = (\bar{m}_{i,j})$ and $M = (m_{i,j})$. Let $\mathbf{v} = M[1]$, $\bar{\mathbf{v}} = \bar{M}[1]$ and $\mathbf{w} = \hat{R}(n, S)$. By construction, we have $\bar{m}_{1,j} = m_{n-a+2,j}$ for any $1 \leq j \leq n$, and $\bar{m}_{i+1,j} = m_{n-t_i+2,j}$ for any $1 \leq j \leq n$ and $1 \leq i \leq \tau$, but also $\bar{\mathbf{v}}[1] = \mathbf{v}[n-a+2]$ and $\bar{\mathbf{v}}[i+1] = \mathbf{v}[n-t_i+2]$ for $1 \leq i \leq \tau$.

Since $a \notin S$, $\hat{R}(n, S)$ must possess a Δ^+ in its a-th component, so that $\bar{\mathbf{v}}[1] = \mathbf{v}[n-a+2] = \mathbf{w}[a] = \Delta^+$ (Lemma 4.4.11). As a consequence, $\bar{\mathbf{v}}$ is a singleton if and only if $\bar{m}_{2,1} = \bar{m}_{3,1} = \ldots = \bar{m}_{\tau+1,1} = 0$, i.e. if and only if $m_{n-t_1+2,1} = m_{n-t_2+2,1} = \ldots = m_{n-t_\tau+2,1} = 0$, which is true by Lemma 4.4.11 if and only if $\mathbf{w}[t_1] = \ldots = \mathbf{w}[t_\tau] = 0$. By definition of $\hat{R}(n, S)$, this holds if and only if $t_1, \dots, t_\tau \in S$, i.e. if and only if $T \subseteq S$. □

Lemma 4.4.13. *Let $T \subseteq S \subseteq \{0, \dots, n-1\}$, $T, S \neq \emptyset$, $a \notin S$. Let $\bar{M}$ be the sub-matrix of M as in Lemma 4.4.12. Then the singleton independence-check procedure is successful on $\bar{M}$ if and only if T is independent from S.*

Proof. By induction on $|T|$.

$|T| = 1$.

Any $T = \{t_1\}$ included in S is obviously independent from S. So we must show that the procedure is always successful in this situation. By Lemma 4.4.12 matrix $\bar{M}$ contains two rows and its first column is a singleton. By removing it and its corresponding row, we remain with a row containing some Δ^+'s, so the procedure is successful.

$|T| = l \quad \implies \quad |T| = l+1.$

Suppose now $T = l+1$. By Lemma 4.4.12 we have a singleton $\bar{M}[1]$. By removing the singleton and its corresponding row, we get a submatrix $\bar{M}'$. By Lemma 4.4.10 we have that T is the shift of $I = J \sqcup \{b\}$, where J is independent from S, $J \subseteq S$ and $b \notin S$. We consider a matrix $\bar{M}'$ as in Lemma 4.4.12. By induction, the procedure is successful on $\bar{M}'$. However, since T is obtained from I by shifting, it means that the rows of $\bar{M}'$ are nothing else that the (same) shift of the rows of $\bar{M}$ (except the row of the first singleton), hence the columns of $\bar{M}$ are a cyclic permutation of the columns of $\bar{M}'$, so that the procedure is successful on $\bar{M}'$ if and only if it is successful on $\bar{M}$. □

Putting all lemmas together and considering $S = S_{C,\alpha}$, we immediately have proved Proposition 4.4.9 and hence Theorem 4.4.8.

Thanks to Theorem 4.4.8, we are able to give an alternative definition of the VW-bound.

Definition 4.4.14. *Let $f_{\mathrm{VW}} \colon \mathcal{E} \to \mathbb{N}$ be the strict border function, defined by:*

$$f_{\mathrm{VW}}(n, S, \mathbf{S}) = \min_{T \in \mathbf{S}} \{\mathrm{rk}(\hat{R}(n, T))\}.$$

*f_{VW} is called the **Van Lint-Wilson (strict border) function**. The (strict border) bound associated to f_{VW} is the **Van Lint-Wilson bound** and it is denoted by δ_{VW}.*

The following corollary explains the link between f_{VW} and the optimal strict border function, b^S, showing that they are the same. This implies that δ_{VW} is sharper than all possible strict border bound (and, of course, also than all strict root bounds).

Corollary 4.4.15. *Let $f_{\mathrm{VW}} \in \mathcal{B}^S$ be the VW function, and $\mathsf{f}^S \in \mathcal{B}^S$ be the optimal strict border function. Then*

$$f_{\mathrm{VW}} = \mathsf{b}^S$$

and δ_{VW} is the optimal strict border bound.

Proof. Given any $(n, S, \mathbf{S}) \in \mathcal{E}'$, it is sufficient to use Theorem 4.3.9, Theorem 3.2.18 and Definition 4.4.14, to obtain:

$$\mathsf{b}^S(n, S, \mathbf{S}) = \min_{T \in \mathbf{S}} \{\mathrm{rk}(M(\hat{R}(n, T)))\} = f_{\mathrm{VW}}(n, S, \mathbf{S})$$

□

Bounding distance using Gröbner bases

This chapter is devoted to bounding the minimum distance of cyclic codes using Gröbner bases. The idea, introduced by Cooper in 1990 [Coo90, Coo91, Coo93] and developed by Chen et al. [CRHT94a, CRHT94b, CRHT94c] is to describe the words of a code as varieties of suitable ideals, and then study them using Gröbner bases. Although this approach was originally proposed to decode cyclic codes up to half of their minimum distance, some authors [ACS90, ACS92, Sal07, MS03, Aug96, Sal02] adapted it also for finding the distance of cyclic codes. We do not deal with the vast area of research regarding the decoding, preferring to focus our attention only on bounding minimum distance. These methods can be roughly divided in two families:

- Newton's identities methods [ACS90, ACS92, MS03, Sal02]

- Power sums methods or Cooper's philosophy [MO09, Sal07, MS03, Sal02].

In Section 5.1 we introduce the notation and necessary backgrounds on Gröbner bases. Section 5.2 explains the methods using power sums , while Section 5.3 contains an overview of the methods using Newton's identities. Our main references for this chapter are [BPW$^+$10, MO09, Cha98, Sal02, Sal07, Aug96].

5.1 Backgrounds

The theory of Gröbner bases was developed by Buchberger [Buc65] in 1965. A useful property is that their computation allows sometimes to solve systems of polynomial equations. In particular, in this subsection we remind the use of Gröbner bases to determine if a system of polynomial equations has solution. Some material is taken from the lecture notes of the course *Coding Theory* lectured by M. Sala and written by D. Frapporti and O. Geil. For a more detailed treatment we refer to [CLO07, Mor05].

Let $\mathbb{K}$ be a field (not necessary finite) , $\overline{\mathbb{K}}$ its algebraic closure. In case $\mathbb{K}$ is finite we write $\mathbb{F}_q$ to indicate the field with q elements, where q is a power of some prime. Let $r \geq 1$ and $R = \mathbb{K}[x_1, \ldots, x_r] = \mathbb{K}[X]$ be a polynomial ring over $\mathbb{K}$ in r variables.

Let $X = \{x_1, \ldots, x_r\}$ be a set of variables. For any $\alpha \in \mathbb{N}^r$ we define a `monomial` X^α:

$$X^\alpha = x_1^{\alpha_1} \ldots x_r^{\alpha_r} \quad \text{with } \alpha = (\alpha_1, \ldots, \alpha_r).$$

We denote by $\mathcal{M} = \mathcal{M}(X) = \{x_1^{\alpha_1} \ldots x_r^{\alpha_r} \mid (\alpha_1 \ldots \alpha_r) \in \mathbb{N}^r\}$ the set of all monomials in the variables $X = (x_1, \ldots, x_r)$.

Definition 5.1.1. *A `monomial ordering` on $\mathbb{K}[X]$ is a binary relation $<$ on $\mathcal{M}(X)$ such that:*

(1) $\forall\ X^\alpha,\ X^\beta \in \mathcal{M},\ X^\alpha \neq X^\beta$, either $X^\alpha < X^\beta$ or $X^\beta < X^\alpha$.

(2) $\forall\ X^\alpha,\ X^\beta,\ X^\gamma \in \mathcal{M}$, if $X^\alpha < X^\beta$, and $X^\beta < X^\gamma$, then $X^\alpha < X^\gamma$.

(3) $\forall\ X^\alpha,\ X^\beta,\ X^\gamma \in \mathcal{M}$, if $X^\alpha < X^\beta$ then $X^\gamma X^\alpha < X^\gamma X^\beta$

(4) $1 < X^\alpha, \forall X^\alpha \in \mathcal{M},\ X^\alpha \neq 1$.

From Definition 5.1.1, we have that a monomial ordering is a well-ordering, i.e. every non-empty subset of $\mathcal{M}$ has a least element. Let $X^\alpha = x_1^{\alpha_1} \ldots x_r^{\alpha_r} \in \mathcal{M}$ and $X^\beta = x_1^{\beta_1} \ldots x_r^{\beta_r} \in \mathcal{M}$, we denote by $\deg(X^\alpha) = \sum_{i=1}^r \alpha_i$ and $\deg(X^\beta) = \sum_{i=1}^r \beta_i$ their total degrees. We provide some examples of monomial orderings.

`Lex.` Lexicographic order induced by $x_r < \cdots < x_1$: $X^\alpha <_{\mathrm{lp}} X^\beta$ if there exists j such that $\alpha_1 = \beta_1, \ldots, \alpha_{j-1} = \beta_{j-1}$, $\alpha_j < \beta_j$.

$$X = (x, y, z), \quad z < y < x \implies xy^5z^3 <_{\mathrm{lp}} x^2yz.$$

`DegLex (or Totlex).` Degree lexicographical order (or total lexicographic order), induced by $x_r < \cdots < x_1$: $X^\alpha <_{\mathrm{D}p} X^\beta$ if either $\deg(X^\alpha) < \deg(X^\beta)$ or $\deg(X^\alpha) = \deg(X^\beta)$ and $X^\alpha <_{\mathrm{l}p} X^\beta$.

`DegRevLex.` Degree reverse lexicographic order induced by $x_r < \cdots < x_1$: $X^\alpha <_{\mathrm{dp}} X^\beta$ if $\deg(X^\alpha) < \deg(X^\beta)$ or $\deg(X^\alpha) = \deg(X^\beta)$ and there exists j such that $\alpha_r = \beta_r, \ldots, \alpha_{j+1} = \beta_{j+1}$, $\alpha_j > \beta_j$.

$$X = (x, y, z), \quad x > y > z \implies xy^4z^3 >_{\mathrm{dp}} x^2y^2z^4.$$

`Block order` Let X and Y be two ordered sets of variables, $<_1$ a monomial order on $\mathbb{K}[X]$ and $<_2$ a monomial order on $\mathbb{K}[Y]$. The block order on $\mathbb{K}[X, Y]$ is the following: $X^{\alpha_1}Y^{\beta_1} < X^{\alpha_2}Y^{\beta_2}$ if $X^{\alpha_1} <_1 X^{\alpha_2}$ or if $X^{\alpha_1} = X^{\alpha_2}$ and $Y^{\beta_1} <_2 Y^{\beta_2}$.

$$X = (x_1, x_2),\ Y = (y_1, y_2, y_3), \quad x_2 <_1 x_1,\ y_3 <_2 y_2 <_2 y_1 \implies x_1^2y_2y_3 < x_1^2y_1^2y_3.$$

Once fixed a monomial order, the following definition is well-posed.

Definition 5.1.2. *Let $<$ be a monomial order on $\mathbb{K}[X]$.Let $f = \sum_\alpha c_\alpha X^\alpha$ be a non-zero polynomial of $\mathbb{K}[X]$, where $c_\alpha \neq 0$. We say that X^β is the* `leading monomial` *of f if $X^\alpha < X^\beta$ for all $\alpha \neq \beta$. We write $\mathrm{LM}(f) = X^\beta$. $\mathrm{LC}(f) = c_\beta$ is called the* `leading coefficient of` *f, $\mathrm{LT}(f) = c_\beta X^\beta$ is called the* `leading term` *of f.*

For any ideal I let $\mathrm{LT}(I)$ be the set of leading terms of element of I, that is $\mathrm{LT}(I) = \{\mathrm{LT}(f) \mid f \in I\}$. We define the `ideal of leading terms` as the ideal generated by the elements of $\mathrm{LT}(I)$. We denote this ideal by $\langle \mathrm{LT}(I) \rangle$.

We can now introduce the definition of Gröbner basis.

Definition 5.1.3. *Let I be an ideal in $\mathbb{K}[X]$. A finite subset $G = \{g_1, \dots, g_m\}$ of I is called a* `Gröbner basis` *for I with respect to the monomial order $<$ if*
$\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(g_1), \dots, \mathrm{LT}(g_m) \rangle$.

Equivalently, G is a Gröbner basis for I if $G \subseteq I$ and if for all $f \in I$ there exists $g_i \in G$ such that $\mathrm{LM}(g_i)$ divides $\mathrm{LM}(f)$. It is easy to see that a Gröbner basis for I is actually a basis of I as an ideal.

Theorem 5.1.4. *For every ideal I in $\mathbb{K}[X]$ and for every monomial ordering $<$ on $\mathcal{M}$, there exists a Gröbner basis G of I.*

Proof. See [Buc06]. □

Moreover, Buchberger provides an effective algorithm ([Buc06, Buc98]) that transforms any finite set of generators of I into a Gröbner basis with respect to $<$.
Many Gröbner bases exist for the same ideal $I \in \mathbb{K}[X]$, but we are interested in a special basis, which is called `reduced`.

Definition 5.1.5. *Let I be an ideal in $\mathbb{K}[X]$. Let G be a Gröbner basis for I with respect to a monomial order $<$. We say that G is* `reduced` *if for all $g \in G$ we have that $\mathrm{LC}(g) = 1$ and for any $g' \in G \setminus \{g\}$ $\mathrm{LT}(g')$ does not divide any monomial of g.*

For an ideal $I \subseteq \mathbb{K}[X]$, $I \neq \{0\}$, the reduced Gröbner basis is unique, so two ideals I_1 and I_2 in are equals if and only if they have the same reduced Gröbner basis. We denote by $G = \mathrm{GB}(I)$ the reduced Gröbner basis of I. Given a Gröbner basis G of an ideal I, we find the reduced Gröbner basis of I by performing successive reductions between the polynomials which compose G. Let $\mathbb{E} \supseteq \mathbb{K}$ be an extension field of $\mathbb{K}$. We denote by $\mathcal{V}_\mathbb{E}(I)$ the `variety of` I `over` $\mathbb{E}$:

$$\mathcal{V}_\mathbb{E}(I) = \{P \in \mathbb{E}^r \mid f(P) = 0 \ \ \forall f \in I\}.$$

The elements of $\mathcal{V}_{\mathbb{E}}(I)$ are sometimes called the $\mathbb{E}$-rational points of I. If $\mathbb{E} = \overline{\mathbb{K}}$, we write $\mathcal{V}(I) = \mathcal{V}_{\mathbb{E}}(I)$ and we say that $\mathcal{V}(I)$ is the `variety of` I. We say that I is `0-dimensional` if $\mathcal{V}(I)$ is finite.

Having a reduced Gröbner basis for an ideal I in $\mathbb{K}[X]$, it is easy to establish if $\mathcal{V}(I) = \emptyset$, as shown below.

Proposition 5.1.6. *Let I be an ideal in $\mathbb{K}[X]$, $G = \mathrm{GB}(I)$ the reduced Gröbner basis of I with respect any monomial order $<$. Then $\mathcal{V}(I) = \emptyset \iff G = \{1\}$.*

Let us suppose to have a system of polynomial equations, $f_1, \dots, f_t \in \mathbb{K}[X]$

$$J = \begin{cases} f_1(X) = 0 \\ \vdots \\ f_t(X) = 0 \end{cases}$$

and we consider the ideal $\mathcal{I}(J)$ generated by the equations: $\mathcal{I}(J) = \langle f_1, \dots, f_t \rangle$. The solution set of J over any extension $\mathbb{E}$ of $\mathbb{K}$ corresponds to the variety of $\mathcal{I}(J)$ over $\mathbb{E}$, i.e.:

$$\begin{aligned} \{P \in \mathbb{E}^r \mid f_1(P) = f_2(P) = \cdots = f_t(P) = 0\} &= \{P \in \mathbb{E}^r \mid f(P) = 0 \;\; \forall f \in I(J)\} \\ &= \mathcal{V}_{\mathbb{E}}(I(J)) \end{aligned}$$

We say that J_t has a solution if there exists $P \in \overline{\mathbb{K}}^r$ such that $f_1(P) = \cdots = f_t(P) = 0$. Clearly, J has a solution if and only if $\mathcal{V}(\mathcal{I}(J)) \neq \emptyset$. Thus, thanks to Proposition 5.1.6, given $G = \mathrm{GB}(I(J))$, the reduced Gröbner basis of $\mathcal{I}(J))$, we have that if $G = \{1\}$, then J has no solution, otherwise it has. From now on, we will speak of ideals and systems interchangeably and, wiyh abuse of notation we will write J for $\mathcal{I}(J)$.

5.2 The Cooper Philosophy

Let $C \in \mathcal{C}_{q,n}$ be a cyclic code, with complete defining set $S_C = \{i_1, \dots .i_{n-k}\}$, with respect to a primitive $n-$th root of unity $\alpha \in \mathbb{F}$, which, from now on, is fixed. We suppose that $c \in C$ is any non-zero word of C and $\mathrm{w}(c) = w \geq 1$. We indicate by $c_{j_1}, \dots, c_{j_w}$ the non-zero components of c, where $0 \leq j_1 < j_2 < \cdots < j_w \leq n-1$, i.e. $c = (c_{j_1}, 0, \dots, 0, c_{j_2}, \dots, c_{j_w}, 0, \dots, 0)$ which corresponds to the polynomial $c(x) = c_{j_1}x^{j_1} + c_{j_2}x^{j_2} + \cdots + c_{j_w}x^{j_w} \in \mathbb{F}_q[x]$. We define $S_i = c(\alpha^i)$ for all $i = \{0, \dots, n-1\}$ and we say that S_i is a `known syndrome` (of c) if $i \in T$, otherwise S_i is called an `unknown syndrome` (of c). Note that if we consider the DFT of c, we have $\mathrm{DFT}(c) = (S_0, S_1, \dots, S_{n-1})$. We have already seen in Section 1.2 that a parity-check matrix for

C is

$$H = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \dots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \dots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{i_{n-k}} & \alpha^{2i_{n-k}} & \dots & \alpha^{(n-1)i_{n-k}} \end{pmatrix}$$

Hence, multiplying Hc^T, we obtain for all $i \in T$:

$$S_i = c(\alpha^i) = c_{j_1}(\alpha^{j_1})^i + \cdots + c_{j_w}(\alpha^{j_w})^i = 0$$

i.e., c is a word of C if and only if all its known syndromes are zero. The $\alpha^{j_1}, \dots, \alpha^{j_w}$ are called the `locations` (of c), and the $c_{j_1}, \dots, c_{j_w}$ are called the `values` (of c). We have some natural constraints which link known syndromes, locations and values:

1. the known syndromes have to be zero:
$$c_{j_1}(\alpha^{j_1})^i + \cdots + c_{j_w}(\alpha^{j_w})^i = 0 \quad \text{for all } i \in T$$
2. the locations are n-th root of unity:
$$(\alpha^{j_i})^n - 1 = 0 \quad \text{for } 1 \le i \le w$$
3. the values belongs to $\mathbb{F}_q$ and are not zero:
$$c_{j_i}^{q-1} - 1 = 0 \quad \text{for } 1 \le i \le w$$

From these constraints we can consider a system, whose variety describes the words of C of weight w. We introduce the variables $z_1, \dots, z_w$, for the locations and the variables $y_1, \dots, y_w$ for the values. Thus, the previous restrictions can be rewritten using these variables:

1. $\sum_{t=1}^{w} y_t z_t^{i_j} = 0$ for $1 \le j \le n-k$
2. $z_i^n - 1 = 0$, for $1 \le i \le w$
3. $y_i^{q-1} - 1 = 0$, for $1 \le i \le w$.

Collecting all these equations in $\mathbb{F}_q[z_1, \dots, z_w, y_1, \dots, y_w]$ in a system, we get:

$$\mathbf{J_C(w)} = \begin{cases} y_1 z_1^{i_1} + \cdots + y_w z_w^{i_1} = 0 \\ \dots \\ y_1 z_1^{i_{n-k}} + \cdots + y_w z_w^{i_{n-k}} = 0 \\ z_1^n - 1 = 0 \\ \dots \\ z_w^n - 1 = 0 \\ y_1^{q-1} - 1 = 0 \\ \dots \\ y_w^{q-1} - 1 = 0 \end{cases} \tag{5.1}$$

We have that any codeword in C of weight w corresponds to a solution of $J_C(w)$. Unfortunately, the converse is not true. The solutions of $J_C(w)$ which do not correspond to any codeword of C are called `spurious` solutions ([Sal02]). In [Sal07] it is proved that a solution $\bar{x} = (\overline{z}_1, \dots, \overline{z}_w, \overline{y}_1, \dots, \overline{y}_w)$ is a spurious solutions of $J_C(w)$ if there are $1 \leq i \neq j \leq w$ such that $\overline{z}_i = \overline{z}_j$ and a refined version of the system $J_C(w)$ is proposed in order to remove all spurious solutions. The new system proposed in [Sal07], $\mathbf{\hat{J}_C(w)}$, is obtained adding to $J_C(w)$, for any $1 \leq i \neq j \leq n$, the polynomials in $\mathbb{F}_q[z_i, z_j]$:

$$p_{i,j} = p(z_i, z_j) = \sum_{h=0}^{n-1} z_i^h z_j^{n-1-h} = \frac{z_i^n - z_j^n}{z_i - z_j},$$

obtaining the following result.

Theorem 5.2.1 ([Sal07, BPW$^+$10]). *Let C be an $[n, k, d]$ cyclic code over $\mathbb{F}_q$ with $(n, q) = 1$ and complete defining set $T = \{i_1, \dots, i_{n-k}\}$. Let $1 \leq w \leq n$ and let $\hat{J}_C(w)$ denote the system:*

$$\begin{cases} y_1 z_1^{i_t} + \dots + y_w z_w^{i_t} = 0, & 1 \leq t \leq n-k \\ z_j^n - 1 = 0, & 1 \leq j \leq w \\ y_j^{q-1} - 1 = 0, & 1 \leq j \leq w \\ p(z_i, z_j) = 0, & 1 \leq i \neq j \leq w \end{cases}.$$

Then, denoting by $A_w(C)$ the number of codewords of weight w in C, and by $\hat{\mathrm{ns}}[w]$ the number of solutions of $\hat{J}_C(w)$, we have: $A_w(C) = \frac{\hat{\mathrm{ns}}[w]}{w!}$. Moreover, for $1 \leq w \leq d$:

- *either $\hat{J}_C(w)$ has no solutions, which is equivalent to $w < d$,*
- *or $\hat{J}_C(w)$ has some solutions, which is equivalent to $w = d$.*

Thanks to Theorem 5.2.1, an algorithm is proposed to compute the minimum distance of a cyclic code, which is an obvious adaption of Proposition 4.2 explained in [Sal02].

Algorithm A

`Input`

A cyclic code $C \in \mathcal{C}_{q,n}$.

A value $w = 1$.

`Output`

The distance $\mathrm{d}(C)$.

```
Cycle
Construct the associated system $\hat{J}_C(w)$.
Compute the Gröbner basis $\hat{G} = \mathrm{GB}(J_C(w))$ of the associated ideal.
If $\hat{G} = \{1\}$ then increase $w$ to $w + 1$

Last step
Output $w$.
```

We provide two examples, applying the Algorithm A to the codes examined in Theorem 4.1.19, showing that their distances are distinct.

Example 5.2.2. We consider the cyclic code C_1 over $\mathbb{F}_q = \mathbb{F}_{3^5}$ of length 11, and defining set $T = \{0, 1, 2, 3, 5\}$. Let us denote with d_1 its distance. By the BCH bound, we have that $d_1 \geq 5$, thus we construct $\hat{J}_{C_1}(5)$ in the polynomial ring $\mathbb{F}_{3^5}[z_1, \ldots, z_5, y_1, \ldots, y_5]$, to check if $d_1 = 5$:

$$\hat{J}_{C_1}(5) = \begin{cases} y_1 + y_2 + y_3 + y_4 + y_5 = 0 \\ y_1 z_1 + y_2 z_2 + y_3 z_3 + y_4 z_4 + y_5 z_5 = 0 \\ y_1 z_1^2 + y_2 z_2^2 + y_3 z_3^2 + y_4 z_4^2 + y_5 z_5^2 = 0 \\ y_1 z_1^3 + y_2 z_2^3 + y_3 z_3^3 + y_4 z_4^3 + y_5 z_5^3 = 0 \\ y_1 z_1^5 + y_2 z_2^5 + y_3 z_3^5 + y_4 z_4^5 + y_5 z_5^5 = 0 \\ z_1^{11} - 1 = 0, \; y_1^{242} - 1 = 0 \\ z_2^{11} - 1 = 0, \; y_2^{242} - 1 = 0 \\ z_3^{11} - 1 = 0, \; y_3^{242} - 1 = 0 \\ z_4^{11} - 1 = 0, \; y_4^{242} - 1 = 0 \\ z_5^{11} - 1 = 0, \; y_5^{242} - 1 = 0 \\ p(z_1, z_2) = 0, \; p(z_1, z_3) = 0, \; p(z_1, z_4) = 0, \\ p(z_1, z_5) = 0, \; p(z_2, z_3) = 0, \; p(z_2, z_4) = 0, \\ p(z_2, z_5) = 0, \; p(z_3, z_4) = 0, \; p(z_3, z_5) = 0, \\ p(z_4, z_5) = 0. \end{cases}$$

We compute its reduced Gröbner basis, $\hat{G}_{C_1}(5)$, with respect to any order, for example `DegRevLex`, with the following variable ordering $z_1 > \cdots > z_5 > y_1 > \cdots > y_5$, to decide if $d_{C_1} = 5$. $\hat{G}_{C_1}(5)$ contains 646 polynomials and we just indicate some elements of $\mathrm{LT}(\hat{G}_{C_1}(5)) = \{y_1, z_1, z_2 y_2, z_2^2 y_3, z_2^2 z_3 y_4, z_4^2 y_2 y_3, y_2^4, z_3^2 y_2^2, z_2^4, z_2^2 y_4^3, \ldots\}$. We have that $\hat{G}_{C_1}(5)$ is different from $\{1\}$, thus a solution exists and therefore $d_1 = 5$.

Example 5.2.3. We consider the cyclic code C_2 over $\mathbb{F}_q = \mathbb{F}_{2^{10}}$ of length 11, and defining set $T = \{0, 1, 2, 3, 5\}$. Let us denote with d_2 its distance. By the BCH bound, we have that $d_2 \geq 5$, thus we construct $\hat{J}_{C_2}(5)$ in the polynomial ring

$\mathbb{F}_{2^{10}}[z_1, \ldots, z_5, y_1, \ldots, y_5]$, to check if $d_2 = 5$:

$$\hat{J}_{C_2}(5) = \begin{cases} y_1 + y_2 + y_3 + y_4 + y_5 = 0 \\ y_1 z_1 + y_2 z_2 + y_3 z_3 + y_4 z_4 + y_5 z_5 = 0 \\ y_1 z_1^2 + y_2 z_2^2 + y_3 z_3^2 + y_4 z_4^2 + y_5 z_5^2 = 0 \\ y_1 z_1^3 + y_2 z_2^3 + y_3 z_3^3 + y_4 z_4^3 + y_5 z_5^3 = 0 \\ y_1 z_1^5 + y_2 z_2^5 + y_3 z_3^5 + y_4 z_4^5 + y_5 z_5^5 = 0 \\ z_1^{11} - 1 = 0, \ y_1^{1023} - 1 = 0 \\ z_2^{11} - 1 = 0, \ y_2^{1023} - 1 = 0 \\ z_3^{11} - 1 = 0, \ y_3^{1023} - 1 = 0 \\ z_4^{11} - 1 = 0, \ y_4^{1023} - 1 = 0 \\ z_5^{11} - 1 = 0, \ y_5^{1023} - 1 = 0 \\ p(z_1, z_2) = 0, \ p(z_1, z_3) = 0, \ p(z_1, z_4) = 0, \\ p(z_1, z_5) = 0, \ p(z_2, z_3) = 0, \ p(z_2, z_4) = 0, \\ p(z_2, z_5) = 0, \ p(z_3, z_4) = 0, \ p(z_3, z_5) = 0, \\ p(z_4, z_5) = 0. \end{cases}$$

Computing its Gröbner basis, $\hat{G}_{C_2}(5)$, with respect `DegRevLex`, with $z_1 > \cdots > z_5 > y_1 > \cdots > y_5$ we obtain, $\hat{G}_{C_2}(5) = \{1\}$, so there are no words of weight 5 in C_2, then $d_2 \geq 6$.

We note that the system $\hat{J}_{C_1}(5)$ and $\hat{J}_{C_2}(5)$ of Example 5.2.2 and Example 5.2.3, respectively, are apparently the same, but the first is defined over $\mathbb{F}_{3^5}[z_1, \ldots, z_5, y_1, \ldots, y_5]$ and the second over $\mathbb{F}_{2^{10}}[z_1, \ldots, z_5, y_1, \ldots, y_5]$. We investigate more in depth this relation between ideals, in Chapter 8, where we call this kind of ideals $\mathcal{F}$-`linked`.

5.3 Newton's Identities

We consider the same settings as the previous section: a cyclic code $C \in \mathcal{C}_{q,n}$, having complete defining set $S_C = \{i_1, \ldots . i_{n-k}\}$ with respect to a fixed primitive $n-$th root of unity $\alpha \in \mathbb{F}$; $c \in C$ any word of C of weight $\mathrm{w}(c) = w$. If $\alpha^{j_1}, \ldots, \alpha^{j_w}$ are the locations of c, we define $X_i = \alpha^{j_i}$, for any $1 \leq i \leq w$. Similarly, if $c_{j_1}, \ldots, c_{j_w}$ are the values of c, we define $Y_i = c_{j_i}$ for any $1 \leq i \leq w$. Following this notation we can rewrite the known and the unknown syndromes of C as $S_j = \sum_{i=1}^{w} X_i^j Y_i$ for any $j \in \{0, \ldots, n-1\}$. The `plain error-locator` polynomial of c is a polynomial in $\mathbb{F}[z]$ defined by:

$$\sigma(z) = \prod_{i=1}^{w}(z - X_i), \tag{5.2}$$

while the classical `error-locator` polynomial is defined (see [ABO09]) by

$$\tilde{\sigma}(z) = \prod_{i=1}^{w}(1 - zX_i).$$

Clearly we have that σ is the reciprocal polynomial of $\tilde{\sigma}$, i.e. $\sigma(z) = z^w \tilde{\sigma}\left(\frac{1}{z}\right)$.

Expanding the product in (5.2) we obtain:

$$\sigma(z) = z^w + \sigma_1 z^{w-1} + \cdots + \sigma_{w-1} z + \sigma_w,$$

where the coefficients $\sigma_1, \ldots, \sigma_w$ are the `elementary symmetric functions of` c, i.e. the elementary symmetric functions of the locations of c with a suitable choice of the sign

$$\sigma_i = (-1)^i \sum_{1 \le j_1 < j_2 < \cdots < j_i \le w} X_{j_1} X_{j_2} \ldots X_{j_i}, \quad 1 \le i \le w.$$

The link between σ and $\tilde{\sigma}$ is also explained in terms of σ_i's, in fact $\tilde{\sigma}(z) = 1 + \sum_{i=1}^{w} \sigma_i z^i$. The elementary symmetric functions of a word c and its syndromes S_i's (or equivalently, its DFT) are linked by the `generalized Newton identities`.

Theorem 5.3.1. *Let $c \in (\mathbb{F}_q)^n$ be a word of weight w, $\mathrm{DFT}(c) = (S_0, \ldots, S_{n-1})$ and $\sigma_1, \ldots, \sigma_w$ the elementary symmetric function of c. Then the following identities hold:*

$$\forall\ i \ge 0, \quad S_{i+w} + \sigma_1 S_{i+w-1} + \cdots + \sigma_w S_i = 0, \tag{5.3}$$

where $S_i = S_{i+n}$.

Proof. See [PW72]. □

Using the generalized Newton identities and the constraints for $\mathrm{DFT}(c)$, in [Aug96] the author presents the following system of equations, where both the S_i's and the σ_i's are the indeterminates, which defines an ideal in $\mathbb{F}_q[S_0, \ldots, S_{n-1}, \sigma_1, \ldots, \sigma_w]$:

$$\mathcal{S}_C(w) = \begin{cases} S_{w+1} + S_w\sigma_1 + \cdots + S_1\sigma_w = 0, \\ S_{w+2} + S_{w+1}\sigma_1 + \cdots + S_2\sigma_w = 0, \\ \vdots \\ S_{n+w} + S_{n+w-1}\sigma_1 + \cdots + S_n\sigma_w = 0, \\ S_{qi \mod n} = S_i^q, \quad 0 \le i \le n-1 \\ S_{i+n} = S_i, \quad 0 \le i \le n-1 \\ S_i = 0, \quad \forall\ i \in S_C \end{cases} \tag{5.4}$$

We give a definition and then summarize the main results which are claimed in [Aug96] concerning $\mathcal{S}_C(w)$.

Definition 5.3.2. *We say that* $(\bar{S}_0, \ldots, \bar{S}_{n-1}) \in (\overline{\mathbb{F}_q})^n$ *(resp.* $(\bar{\sigma}_1, \ldots, \bar{\sigma}_w)$*) is a* ***truncated solution*** *of* $\mathcal{S}_C(w)$ *if there exist* $(\bar{\sigma}_1, \ldots, \bar{\sigma}_w) \in (\overline{\mathbb{F}_q})^w$ *(resp.* $(\bar{S}_0, \ldots, \bar{S}_{n-1})$*) such that* $(\bar{S}_0, \ldots, \bar{S}_{n-1}, \bar{\sigma}_1, \ldots, \bar{\sigma}_w)$ *is a solution of* $\mathcal{S}_C(w)$*. In this case* $(\bar{\sigma}_1, \ldots, \bar{\sigma}_w)$ *is called an* ***extended solution*** *corresponding to* $(\bar{S}_0, \ldots, \bar{S}_{n-1})$*.*

We remark that what we defined as `truncated solution` is simply called solution in [Aug96]. In the next theorem we use the notation in term of σ rather than in terms of $\tilde{\sigma}$.

Theorem 5.3.3. *Let* C *be an* $[n, k, d]$ *cyclic code over* $\mathbb{F}_q$ *with defining set* S_C*. Then we have the following properties.*

(i) The n-tuples $(\bar{S}_0, \ldots, \bar{S}_{n-1}) \in (\overline{\mathbb{F}_q})^n$ *which are truncated solutions of* $\mathcal{S}_C(w)$ *are the* DFT *of the codewords of weight less than or equal to* w*.*

(ii) Let $(\bar{S}_0, \ldots, \bar{S}_{n-1}) \in (\overline{\mathbb{F}_q})^n$ *be a truncated solution of* $\mathcal{S}_C(w)$ *and* c *be the codeword of weight* $w_0 \leq w$ *with* $\mathrm{DFT}(c) = (\bar{S}_0, \ldots, \bar{S}_{n-1})$*. Let* $\sigma_c(z)$ *be the plain locator polynomial of* c*. Then the set of extended solutions corresponding to* $(\bar{S}_0, \ldots, \bar{S}_{n-1})$ *is*

$$\mathcal{F}' = \left\{ (\bar{\sigma}_1, \ldots, \bar{\sigma}_w) \in (\overline{\mathbb{F}_q})^w \mid \sigma_c(z) \textit{ divides } (z^w + \sum_{i=1}^{w} \sigma_i z^{w-i}) \right\}.$$

(iii) The number of solution of $\mathcal{S}_C(d)$ *is finite. Each truncated solution* $(\bar{S}_0, \ldots, \bar{S}_{n-1})$ *is the* DFT *of a minimum weight codeword. Each truncated solution* $(\bar{\sigma}_1, \ldots, \bar{\sigma}_w)$ *is the set of coefficients of the plain error locator polynomial of a minimum weight codeword.*

Proof. See [Aug96, Cha98]. □

We believe that the result of Theorem 5.3.3 is true, assuming that some conditions are added to $\mathcal{S}_C(w)$, in order to avoid that $(\bar{S}_0, \ldots \bar{S}_{n-1}) = (0, \ldots, 0)$ is a truncated solution. We call $\hat{\mathcal{S}}_C(w)$ the system obtained adding these conditions to $\mathcal{S}_C(w)$.

The important consequence is that given a cyclic code C such that there is not any word of weight less than w, if $\hat{\mathcal{S}}_C(w)$ has solutions, then the distance of C is w. Thus, we can proposed an algorithm analogous to Algorithm A of the previous section, which formalizes the approach of [Aug96] in its example of Section 4.1 .

Algorithm B

`Input`

A cyclic code $C \in \mathcal{C}_{q,n}$.

A value $w = 1$.

`Output`
The distance $\mathrm{d}(C)$.

`Cycle`
Construct the associated system $\hat{\mathcal{S}}_C(w)$.
Compute the Gröbner basis $\hat{G} = \mathrm{GB}(\hat{\mathcal{S}}_C(w))$ of the associated ideal.
If $\hat{G} = \{1\}$ then increase w to $w + 1$

`Last step`
Output w.

If we want to keep use $\mathcal{S}_C(w)$ rather than $\hat{\mathcal{S}}_C(w)$, we provide an alternative algorithm.

Algorithm C

`Input`
A cyclic code $C \in \mathcal{C}_{q,n}$.
A value $w = 1$.

`Output`
The distance, $\mathrm{d}(C)$.

`Cycle`
Construct the associated system $\mathcal{S}_C(w)$.
Compute the Gröbner basis $G = \mathrm{GB}(\mathcal{S}_C(w))$ of the associated ideal.
If $\{S_0, \dots, S_{n-1}\} \subseteq G$ then increase w to $w + 1$

`Last step`
Output w.

Part II

Main results

A New Bound

In this chapter we use Theorem 2.2.16 and the singleton procedure in order to prove a bound, called bound `C`, which is a simultaneous generalization of the Hartmann-Tzeng bound and of the BS bound. Bound `C` has a computational complexity slightly larger than that of the Roos bound. It turns out from extensive computations that bound `C` is often tighter than any other known root bound (including the Roos bound). This result was preliminary presented in [PS13] and solves an open problem proposed in [BS07]. From now on, during this chapter we adopt the notation used in Chapter 3.2. In particular, we fix α a primitive n-th root of unity over $\mathbb{F}_q$ and we write $S_C = S_{C,\alpha}$.

The main result in this chapter is Theorem 6.1.13. We postpone its statement because first we need two prove two special cases (Bound I and Bound II) presented below and whose proofs are given in Section 6.1.

Proposition 6.0.4 (Bound I)**.** *Let C be an $\mathbb{F}_q[n,k,d]$ cyclic code with defining set S_C and $(q,n) = 1$. Suppose that there are ℓ, m, r, $s \in \mathbb{N}$, $1 \leq m \leq \ell$ and $i_0 \in \{ 0, \dots, n-1 \}$ such that:*

a) $(i_0 + j)_n \in S_C$, $\forall j = 0, \dots, \ell - 1$,

b) $(i_0 + j)_n \in S_C$,

$$\forall j = i_0 + \ell + r + h(m+r) + 1, \dots, \; i_0 + \ell + r + m + h(m+r)$$
$$\forall 0 \leq h \leq s-1$$

Then

- *if* $(m + r, n) \leq m$*:*

$$d \geq \ell + 1 + s - r \left\lfloor \frac{\ell}{m+r} \right\rfloor - \max \{ (\ell)_{m+r} - m, 0 \} ; \tag{6.1}$$

- *otherwise*

$$d \geq \ell + 1. \tag{6.2}$$

The above statement is expressed in classical notation and seems extremely complicated. However it is a natural generalization of known bounds, as it is immediate once it is expressed in $\mathcal{U}$ notation.

Proposition 6.0.5 (Bound I)**.** *Let C be an $[n,k,d]$ cyclic code with defining set S_C. Suppose that there are $\ell, s, m, r, \rho \in \mathbb{N}$, $\ell \geq m \geq 1$, $s \geq 1$, $\rho \geq 1$, $r \geq 1$ such that*

$$((0^\ell)(\Delta^r))((0^m)(\Delta^r))^s \preccurlyeq R(n, S_C)^\rho. \tag{6.3}$$

Then

- *if $(m+r, n) \leq m$:*
$$d \geq \ell + 1 + s - r \left\lfloor \frac{\ell}{m+r} \right\rfloor - \max\{\, (\ell)_{m+r} - m, 0 \,\}; \tag{6.4}$$
- *otherwise*
$$d \geq \ell + 1. \tag{6.5}$$

Corollary 6.0.6. *In Proposition 6.0.5 we can substitute condition* (6.3) *with*

$$((\Delta^r)(0^m))^s((\Delta^r)(0^\ell)) \preccurlyeq R(n, S_C)^\rho.$$

Proof. See Lemma 3.2.11. □

Remark 6.0.7. We can see Proposition 6.0.4 as a generalization of the HT bound. In fact with $\ell = m$ the statement of Proposition 6.0.5-(6.4) reduces to Definition 3.4.13 and Corollary 3.4.15.

We claim another bound, similar to bound I:

Proposition 6.0.8 (Bound II)**.** *Let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$ with defining set S_C. Suppose that there are $\lambda, \mu, s \in \mathbb{N}$, $\lambda \geq 1$, $\mu \geq 2$, $s \geq \lambda + 1$, $(n, \mu) \leq \mu - 1$, $i_0 \in \{0, \ldots, n-1\}$ such that:*

a) $(i_0 + j)_n \in S_C$, $j = 0, \ldots, \lambda\mu - 1$,

b) $(i_0 + j)_n \in S_C$, $j = (\lambda + h)\mu + 1, \ldots, (\lambda + h)\mu + \mu - 1$, $0 \leq h \leq s - 1$,

Then:

- *if $(n, \mu) \leq \mu - 1$:*
$$d \geq \lambda\mu + \mu + s - \lambda - 1;$$
- *otherwise if $\mu \mid n$:*
$$d \geq \lambda\mu + \mu.$$

Again, the $\mathcal{U}$ notation is more clear, as follows.

Proposition 6.0.9. *Let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$ with defining set S_C. Suppose that there are $\lambda,\mu,s\in\mathbb{N}$, $\lambda\geq 1$, $\mu\geq 2$, $s\geq\lambda+1$ such that:*

$$(0^{\mu\lambda}\Delta)(0^{\mu-1}\Delta)^s \preccurlyeq R(n,S_C)^\rho. \tag{6.6}$$

Then:

- *if $(n,\mu)\leq\mu-1$:*
$$d\geq\lambda\mu+\mu+s-\lambda-1; \tag{6.7}$$
- *otherwise if $\mu\mid n$:*
$$d\geq\lambda\mu+\mu. \tag{6.8}$$

Corollary 6.0.10. *In Proposition 6.0.9 we can substitute condition* (6.6) *with*

$$(\Delta 0^{\mu-1})^s(\Delta 0^{\mu\lambda}) \preccurlyeq R(n,S_C)^\rho.$$

Proof. See Lemma 3.2.11. □

Remark 6.0.11. Proposition 6.0.9 is clearly a generalization of BS bound (see Definition 3.4.34 and Corollary 3.4.36), and for the rare cases in which $\mu|n$, it is exactly the BS bound.

Remark 6.0.12. We note that bound II, when applicable, is sharper than bound I. In fact, if $(0^{\mu\lambda}\Delta)(0^{\mu-1}\Delta)^s \preccurlyeq R(n,S_C)^\rho$ for $\mu\geq 2$, $s\geq\lambda+1$, in notation of Proposition 6.0.5 it means $(0^\ell\Delta^r)(0^m\Delta^r)^s \preccurlyeq R(n,S_C)^\rho$ with $\ell=\mu\lambda$, $r=1$, $m=\mu-1$ and then Proposition 6.0.5 gives a value d_I

$$d_I\geq\mu\lambda+1+s-\left\lfloor\frac{\mu\lambda}{\mu}\right\rfloor-\max\{\,(\mu\lambda)_\mu-(\mu-1),0\,\}=\mu\lambda+1+s-\lambda$$

while Proposition 6.0.9 gives a value d_{II}

$$d_{II}\geq\mu\lambda+\mu+s-\lambda-1$$

and since $\mu\geq 2$ then $d_{II}\geq d_I$.

6.1 Proofs of bound I and bound II

In this section we provide the proofs of Proposition 6.0.5, and Proposition 6.0.9.

Remark 6.1.1. The main tool we use to prove Proposition 6.0.5 is Theorem 2.2.16 which, in principle, allows us to work only with matrices that have as entries just 0 or $\Delta^{\!\!+}$. Nevertheless during the proof we use matrices that have also Δ as entry. A Δ can be either 0 or $\Delta^{\!\!+}$, the correctness of the proof is not affected by either choice.

Proof. (Proposition 6.0.5) The general plan of the proof is as follows. Thanks to Theorem 2.2.16 we aim at proving that

$$\min\{\,\mathrm{rk}(M(\mathbf{v})) \mid \mathbf{v}\in\mathcal{A}(R(n,S_C))\,\} \geq \ell+1+s-r\left\lfloor\frac{\ell}{m+r}\right\rfloor-\max\{\,(\ell)_{m+r}-m,0\,\}\,.$$

In order to do that, for any $\mathbf{v}\in\mathcal{A}(n,S_C)$, we need to choose $\ell+s+1$ rows in $M(\mathbf{v})$ and we must prove that, discarding at most $r\left\lfloor\frac{\ell}{m+r}\right\rfloor+\max\{\,(\ell)_{m+r}-m,0\,\}$ rows, we actually obtain a set of rows for which the singleton procedure is successful.

We can suppose w.l.o.g. that $i_0=n-\ell$ (see Lemma 3.2.9), so that:

$$\mathbf{v}=\underbrace{\Delta\ldots\Delta}_{r}(\underbrace{0\ldots0}_{m}\underbrace{\Delta\ldots\Delta}_{r})^s\ldots\underbrace{0\ldots0\ldots0}_{\ell}.$$

From now on, the meaning of $\mathbf{v}$ is fixed. Let i' be the primary pivot of $\mathbf{v}$ (see Definition 3.4.16). We can suppose that $1\leq i'\leq r$, otherwise $\mathbf{v}=0^r(0^m\Delta^r)^s\ldots0^\ell$ and so $(0^{\ell+r+m}\Delta^r)(0^m\Delta^r)^{s-1}\preccurlyeq\mathbf{v}$ (Definition 3.3.8) and the bound would be trivially satisfied, since it would give:

$$\begin{aligned} d &\geq \ell+r+m+1+s-1-\left\lfloor\frac{\ell+r+m}{m+r}\right\rfloor r-\max\{\,(\ell+m+r)_{m+r}-m,0\,\}\\ &=\ell+r+m+s-\left\lfloor\frac{\ell}{m+r}\right\rfloor r-\max\{\,(\ell)_{m+r}-m,0\,\}\\ &\geq \ell+r+1+s-\left\lfloor\frac{\ell}{m+r}\right\rfloor r-\max\{\,(\ell)_{m+r}-m,0\,\}\,. \end{aligned}$$

Let i'' be the secondary pivot of $\mathbf{v}$ with respect to the block $(0^m\Delta^r)^s$ (see Definition 3.4.17). We can suppose $s(m+r)+r+1\leq i''\leq s(m+r)+r+m$, otherwise we have $(0^\ell\Delta^r)(0^m\Delta^r)^{s+1}\preccurlyeq\mathbf{v}$ and the bound is trivially satisfied:

$$\begin{aligned} d &\geq \ell+1+s+1-\left\lfloor\frac{\ell}{m+r}\right\rfloor r-\max\{\,(\ell+m+r)_{m+r}-m,0\,\}\\ &\geq \ell+1+s-\left\lfloor\frac{\ell}{m+r}\right\rfloor r-\max\{\,(\ell)_{m+r}-m,0\,\}\,. \end{aligned}$$

We note that $\mathbf{v}[i''-z\cdot(m+r)]=0$ for any $z=1,\ldots,s$. Moreover, i' and i'' may coincide, but this is not a problem.

Now, we are going to choose $(\ell+1+s)$ rows of $M(\mathbf{v})$. We start from the $((n-i'+k)_n+1)-$th rows with $k=1,\ldots,m$, that is, we take the row with the primary pivot in the first position and its shifts up to the $(m-1)-$th shift included. We collect these rows in submatrix T_1 and we note that they are clearly linearly independent, applying the singleton procedure.

$$T_1 = \left(\begin{array}{cccccccccccccccccccccc}
\Delta^{+} & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta^{+} & \ldots & \ldots & \ldots & 0 & \ldots & \ldots & 0 \\
0 & \Delta^{+} & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta^{+} & \ldots & \ldots & \ldots & 0 & \ldots & 0 \\
0 & 0 & \Delta^{+} & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta^{+} & \ldots & \ldots & \ldots & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \ldots & 0 & \Delta^{+} & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta^{+} & \ldots & \ldots & 0 & \ldots \\
 & & & \downarrow & & & & & & & & & & & & & & & & & & \\
 & & & m & & & & & & & & & & & & & & & & & &
\end{array}\right)$$

We now consider the $(k+1)$-th rows for $k = m, \ldots, \ell$, collected in submatrix T_2.

$$T_2 = \left(\begin{array}{cccccccccccccccccccccc}
0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & \Delta^{+} & \ldots & \ldots & \ldots & \ldots & 0 & \ldots & \ldots \\
0 & \ldots & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & \Delta^{+} & \ldots & \ldots & \ldots & \ldots & 0 & \ldots \\
0 & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & \Delta^{+} & \ldots & \ldots & \ldots & \ldots & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \ldots & \ldots & 0 & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & \Delta^{+} & \ldots & \ldots \\
0 & \ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & 0 & \ldots & 0 & \Delta & \ldots & \Delta & \ldots & \Delta^{+} & \ldots \\
 & & \downarrow & & & & & & \downarrow & & & & & & & & & & & & & \\
 & & m & & & & & & \ell & & & & & & & & & & & & &
\end{array}\right)$$

Note that T_1 and T_2 have no common rows. Note also that in T_2 for any row $h = 1, \ldots, \ell + 1 - m$ and any column $1 \leq j \leq (s-1)(m+r) + m$ we have:

$$T_2[h, j] = \Delta \implies T_2[h, j + (m+r)] = \Delta \tag{6.9}$$

Moreover, T_2 has full rank as the following lemma shows.

Lemma 6.1.2. *The singleton procedure is successful for T_2 and thus* $\mathrm{rk}(T_2) = \ell - m + 1$.

Proof. We are going to prove that the singleton procedure is successful for all the rows of T_2. We have that $\mathbf{v}[i'] = \Delta^{+}$ and $\mathbf{v}[i] = 0,\ \forall\ i \in \{\, i'-1, \ldots, i'-\ell \,\}$. In particular $\mathbf{v}[i] = 0,\ \forall\ i \in \{\, i'-1, \ldots, i'-\ell+m \,\}$.

We note that since every row of T_2 is obtained from a right-shift of the previous one and the first row of T_2 is obtained shifting $\mathbf{v}$ of m positions to the right, so for $1 \leq h \leq \ell - m - 2$ it holds

$$T_2[h+1, j] = T_2[h, j-1] \qquad \text{and} \qquad T_2[1, j] = \mathbf{v}[j - m].$$

At the first step we s-delete the first row and the $(i'+m)-$th column, since $T_2[i'+m]$ is a singleton, in fact for $2 \leq h \leq \ell - m + 1$:

$$T_2[h, i'+m] = T_2[1, i'+m-(h-1)] = \mathbf{v}[i'-(h-1)] = 0$$

while $T_2[1, i'+m] = \mathbf{v}[i'] = \Delta^{+}$.

Suppose now we have s-deleted the first j rows, we want to show that the matrix $T_2^{(j)}$ obtained from these j s-deletions has a singleton in $T_2^{(j)}[i'+m+j]$. In fact, for $2 \leq h \leq \ell - m + 1 - j$:

$$\begin{aligned}
T_2^{(j)}[h, (i'+m+j)] &= T_2[j+h, (i'+m+j)] \\
&= T_2[1, i'+m-(h-1)] \\
&= \mathbf{v}[i'-(h-1)] = 0
\end{aligned}$$

while $T_2^{(j)}[1,(i'{+}m{+}j)] = T_2[j{+}1,(i'{+}m{+}j)] = T_2[1,i'{+}m] = \mathbf{v}[i'] = \Delta^{+}$. After $(\ell{-}m)$ steps we have that $T_2^{(\ell-m)}$ is the last row of the matrix T_2, (i.e. $T_2^{(\ell-m)} = T_2[\ell{-}m{+}1]$), which is different from zero, since $T_2[\ell{-}m{+}1, i'{+}\ell{+}1] = T_2[1,i'{+}m] = \mathbf{v}[i'] = \Delta^{+}$. □

Since all the rows of T_2 have a block of zeros in the first m-positions, they are linearly independent from all the rows in T_1. We can conclude that any matrix containing T_1 and T_2 has rank at least $\ell+1$, obtaining (6.5). If $(m+r,n) \le m$ we can also consider a third and last submatrix, T_3, formed by the $((n{-}r{-}k{\cdot}(m{+}r))_n{+}1)-$th rows, for $k = 0,\dots,(s-1)$:

$$T_3 = \left(\begin{array}{cccccccccccccccccccccc}
0 & \cdots & 0 & \Delta & \cdots & \Delta & \cdots & 0 & \cdots & 0 & \Delta & \cdots & \Delta & 0 & \cdots & 0 & \Delta & \cdots & \Delta & \cdots & \Delta^{+} & \cdots \\
0 & \cdots & 0 & \Delta & \cdots & \Delta & \cdots & 0 & \cdots & 0 & \Delta & \cdots & \Delta & \cdots & \Delta^{+} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & \Delta & \cdots & \Delta & \cdots & \Delta^{+} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & \downarrow & & & \downarrow & & \downarrow & & & & & & & & & & & & & \downarrow & \\
 & & m & & & m{+}r & & i''{-}r{-}(s{-}1)(m{+}r) & & & & & & & & & & & & & i''{-}r &
\end{array}\right)$$

Lemma 6.1.3. *The singleton procedure is successful for T_3 and thus $\mathrm{rk}(T_3) = s$. Moreover the $[k]$-th row of T_3 is the row corresponding to the singleton $T_3[i'' - r - (k-1)(m+r)]$ for $k = s, s-1, \dots, 1$.*

Proof. We note that the rows of T_3, by construction, have the property that $T_3[a{+}1,h] = T_3[a,h{+}(m{+}r)]$ because each row is a $(m{+}r)$ left shift of the previous one. This is sufficient to prove that $T_3(i'' - r - (s-1)(m+r))$ is a singleton. We claim that the $s-$th row of T_3 corresponds to a singleton. Indeed

$$T_3[s,i''{-}r{-}(s{-}1)(m{+}r)] = T_3[1,i''{-}r{-}(s{-}1)(m{+}r){+}(s{-}1)(m{+}r)] = T_3[1,i''{-}r] = \Delta^{+}$$

and for $k = 1,\dots,s-1$:

$$T_3[k,i''{-}r{-}(s{-}1)(m{+}r)] = T_3[1,i''{-}r{-}(s{-}1)(m{+}r){+}(k{-}1)(m{+}r)] = T_3[i''{-}r{-}(s{-}k)(m{+}r)] = 0$$

so we can s-delete it. Once this is done, we might also s-delete the $(s-1)-$th row, since

$$T_3[s{-}1,i''{-}r{-}(s{-}2)(m{+}r)] = T_3[1,i''{-}r{-}(s{-}2)(m{+}r){+}(s{-}2)(m{+}r)] = T_3[1,i''{-}r] = \Delta^{+}$$

and for $k = 1,\dots,s-2$:

$$T_3[k,i''{-}r{-}(s{-}2)(m{+}r)] = T_3[i''{-}r{-}(s{-}2)(m{+}r){+}(k{-}1)(m{+}r)] = T_3[1,i''{-}r{-}(s{-}1{-}k)(m{+}r)] = 0.$$

In this way for any row of T_3 we obtain a singleton in $T_3\,[i'' - r - k(m+r)]$ for $k = 0,\dots,s-1$, by recursively s-deleting from the last row to the first. □

Collecting all these submatrices T_1, T_2, T_3, we obtain an $(\ell+1+s) \times n$ matrix T, as follows:

$$T = \left(\begin{array}{cccccccccccccccccccccccccc}
\Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & 0 & \dots & \dots & 0 & \rightarrow & 1 & \\
0 & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & 0 & \dots & 0 & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & \mathbf{T_1} \\
0 & 0 & \Delta^{+} & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & 0 & \dots & & & \\
\hline
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & 0 & \dots & \dots & \rightarrow & m+1 & \\
0 & \dots & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & 0 & \dots & & & \\
0 & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & & & \mathbf{T_2} \\
0 & \dots & 0 & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & & & \\
0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \rightarrow & \ell+1 & \\
\hline
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & & & \\
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & & & \mathbf{T_3} \\
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \rightarrow & \ell+1+s & \\
 & & \downarrow & & & \downarrow & & & & & & & & & & & & & & & & & & & \\
 & & m & & & m+r & & & & & & & & & & & & & & & & & & &
\end{array}\right)$$

Observe that the rows from $(m+1)$ to $(\ell+s+1)$ have a block of zero in the first m positions, so we can obviously s-delete the first m rows (i.e the rows of T_1). After these first m s-deletions we obtain a matrix T' composed of the last $(\ell+1+s-m)$ rows of T, as the following:

$$T' = \left(\begin{array}{cccccccccccccccccccccccc}
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & 0 & \dots & \dots & \rightarrow & m+1 \\
0 & \dots & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & 0 & \dots & & \\
0 & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & & \\
0 & \dots & 0 & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & & \\
0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \rightarrow & \ell+1 \\
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & & \\
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & & \\
0 & \dots & 0 & \Delta & \dots & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \rightarrow & \ell+1+s \\
 & & \downarrow & & & \downarrow & & & & & & & & & & & & & \downarrow & & \downarrow & & & \\
 & & m & & & m+r & & & & & & & & & & & & & s(m+r) & & i''-r & & &
\end{array}\right)$$

where $1 + s(m+r) \leq i'' - r \leq m + s(m+r)$ by hypothesis. We note that T' is composed by the rows of T_2 and T_3.

We use the singletons of T_3 to proceed with the singleton procedure, but in order to do that we have to discard some rows in T_2. More precisely, let us define:

$$B_k = \{\, h \mid T_2[h, i'' - r - k(m+r)] = \Delta \,\} \qquad \text{for } k = 0, \dots, s-1$$

then the rows to discard in T_2 in order that $T[i'' - r - k(m+r)]$ becomes a singleton for $k = 0, \dots, s-1$ are:

$$\mathbf{B} = \cup_{k=0}^{s-1} B_k. \tag{6.10}$$

Lemma 6.1.4. *Let $0 \leq k < k' \leq s-1$, then $B_{k'} \subseteq B_k$.*

Proof. It follows directly from (6.9). □

Corollary 6.1.5. $\mathbf{B} = B_0 = \{\, h \mid T_2[h, i'' - r] = \Delta \,\}$.

Thanks to Corollary 6.1.5, since $s(m+r)+1 \leq i'' - r \leq s(m+r)+m$, if we define $\eta_j = |\,\{\, h \mid T_2[h, s(m+r)+j] = \Delta \,\}\,|$, we have:

$$|\mathbf{B}| \leq \max\{\, \eta_j \mid 1 \leq j \leq m \,\}\,.$$

and we can further improve this result with the following lemma, which is not difficult to prove.

Lemma 6.1.6. *For* $1 \leq j \leq m$*:*

$$\eta_1 \geq \eta_2 \geq \cdots \geq \eta_m.$$

Thanks to lemma 6.1.6 we are able to estimate the maximal number of rows of T_2 that we have to discard.

Lemma 6.1.7.

$$|\mathbf{B}| \leq \eta_1 \leq \left\lfloor \frac{\ell}{m+r} \right\rfloor r + \max\{\, (\ell)_{m+r} - m, 0 \,\}$$

Proof. For Corollary 6.1.5 and Lemma 6.1.6 we have $|\mathbf{B}| \leq \eta_1$. Now:

$$\eta_1 = |\,\{\, h \mid T_2[h, s(m+r)+1] = \Delta \,\}\,|, \text{ but recall } 1 \leq h \leq \ell + 1 - m.$$

We rewrite $\mathbf{v}$ in the worst case where $i'' = s(m+r)+r+1$:

$$\begin{array}{cccccccccccccccccccc}
\mathbf{v}= & \Delta & \ldots & \Delta & 0 & \ldots & 0 & (\Delta^r 0^m)^{s-2} & \Delta & \ldots & \Delta & 0 & & \ldots & 0 & \Delta & \ldots & \Delta & \Delta^{+} & \ldots\ \ldots \\
 & \downarrow & & \downarrow & & & \downarrow & & & & & & \downarrow & & \downarrow & & & & \downarrow & \\
 & 1 & & r & & & m{+}r & & & & & & s(m{+}r){-}m{+}1 & & s(m{+}r) & & & & s(m{+}r){+}r{+}1 &
\end{array}$$

Since $T_2[1, s(m+r)+1] = \mathbf{v}[s(m+r)+1-m] = 0$, we have

$$\begin{aligned}
\eta_1 &= |\,\{\, h \mid T_2[h, s(m+r)+1] = \Delta, 1 \leq h \leq \ell+1-m \,\}\,| \\
&= |\,\{\, h \mid T_2[h, s(m+r)+1] = \Delta, 2 \leq h \leq \ell+1-m \,\}\,|.
\end{aligned}$$

Now $T_2[h+1, j] = T_2[h, j-1]$ (for $h \geq 1$) and $T_2[1, j] = \mathbf{v}[j-m]$, by construction of T_2. So:

$$\begin{aligned}
\eta_1 &= |\,\{\, h \mid T_2[h, s(m+r)+1] = \Delta, 2 \leq h \leq \ell+1-m \,\}\,| \\
&= |\,\{\, h \mid T_2[1, s(m+r)+1-(h-1)] = \Delta, 2 \leq h \leq \ell+1-m \,\}\,| \\
&= |\,\{\, h \mid \mathbf{v}[s(m+r)-m+2-h] = \Delta, 2 \leq h \leq \ell+1-m \,\}\,| \\
&= |\,\{\, h \mid \mathbf{v}[s(m+r)+2-h] = \Delta, 2 \leq h \leq \ell+1 \,\}\,|
\end{aligned}$$

Thus, to compute η_1 we have to count the number of Δ's we encounter, from $\mathbf{v}[s(m+r)]$ to $\mathbf{v}[s(m+r)-\ell+1]$ (i.e. from $\mathbf{v}[s(m+r)]$ and going back ℓ positions). Let us consider the worst case, which is when $\ell \leq s(m+r)$. Passing through the block $(0^m\Delta^r)^s$ from right to left through ℓ positions, every $m+r$ steps we meet a block formed by r Δ's and m 0's, thus the contribution to η_1 per block is r. Since we move by ℓ positions only, we cross no more than $\left\lfloor \frac{\ell}{m+r} \right\rfloor$ such blocks and so we have $\eta_1 \leq \left\lfloor \frac{\ell}{m+r} \right\rfloor r + \eta_1'$, where η_1' are the Δ's coming from the last $(\ell)_{m+r}$ steps left. The first m-positions we meet doing the last $(\ell)_{m+r}$ steps are zero, since they correspond to the last block $(\Delta^r 0^m)$, thus η_1' can be at most $(\ell)_{m+r} - m$ and it is non-negative only if $(\ell)_{m+r} \geq m$. In conclusion: $\eta_1 \leq \left\lfloor \frac{\ell}{m+r} \right\rfloor r + \max\{\,(\ell)_{m+r} - m, 0\,\}$. □

Thanks to Lemma 6.1.7, discarding at most $\left\lfloor \frac{\ell}{m+r} \right\rfloor r + \max\{\,(\ell)_{m+r} - m, 0\,\}$ rows of T_2, we can remove by s-deletions T_3 from T'.The matrix that remains is a submatrix $\widetilde{T}$ of T_2 not having row indeces in $\mathbf{B}$. Note that $\widetilde{T}$ has full rank, because T_2 has full rank by Lemma 6.1.2. So we have proved Proposition 6.0.5. □

Example 6.1.8. Let C be a cyclic code of length n, with defining set S_C satisfying the assumptions of Proposition 6.0.5 with parameters $\ell = 7$, $m = 2$, $r = 1$, $s = 5$. We want to prove that by Proposition 6.0.5 the distance of the code C is at least $d \geq 7 + 1 + 5 - \left\lfloor \frac{7}{2+1} \right\rfloor 1 - \max\left\{\,(7)_{3+2} - 2, 0\,\right\} = 11$. Let $\mathbf{v} \in \mathcal{A}(R(n, S_C))$ with $\mathbf{v}[1] = \Delta^{\!+}$. The matrix T is:

$$\left(\begin{array}{ccccccccccccccccccccccccccc}
\Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & \Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
\hline
0 & 0 & \Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta^{\!+} & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \dots & \dots \\
\hline
0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & \Delta & 0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & \Delta & 0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots \\
0 & 0 & \Delta & \Delta^{\!+} & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \dots & \dots
\end{array}\right)$$

For the secondary pivot we have two possibilities: $i'' = 11$ or $i'' = 12$. We show that in both cases it is possible to obtain 11 s-deletions, removing at most $\left\lfloor \frac{7}{2+1} \right\rfloor 1 + \max\left\{\,(7)_{3+2} - 2, 0\,\right\} = 2$ rows from the matrix T.

Case 1: $i'' = 11$.

$\Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 1-st s-deletion
$0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 2-nd s-deletion
$0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 8-th s-deletion
~~$0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots$~~ $\to$ **REMOVED**
$0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 9-th s-deletion
$0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta \dots \to$ 10-th s-deletion
~~$0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta \dots$~~ $\to$ **REMOVED**
$0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta \dots \to$ 11-th s-deletion
$0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 7-th s-deletion
$0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 6-th s-deletion
$0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 5-th s-deletion
$0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 4-th s-deletion
$0\ 0\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 3-rd s-deletion

Case 2: $i'' = 12$.

$\Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 1-st s-deletion
$0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 2-nd s-deletion
$0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 8-th s-deletion
$0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 9-th s-deletion
~~$0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta \dots$~~ $\to$ **REMOVED**
$0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta \dots \to$ 10-th s-deletion
$0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta \dots \to$ 11-th s-deletion
~~$0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+} \dots$~~ $\to$ **REMOVED**
$0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 7-th s-deletion
$0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 6-th s-deletion
$0\ 0\ \Delta\ 0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 5-th s-deletion
$0\ 0\ \Delta\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 4-th s-deletion
$0\ 0\ \Delta\ \Delta\ \Delta^{+}\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta\ \Delta \dots \to$ 3-th s-deletion

In a similar way we prove Proposition 6.0.9.

Proof. (of Proposition 6.0.9) We can suppose w.l.o.g. that $i_0 = n-\lambda\mu$ (see Lemma 3.2.9), so that $\mathbf{v} = \Delta\overbrace{\underbrace{0\dots0\Delta}_{\mu}\dots\dots\underbrace{0\dots0\Delta}_{\mu}}^{s-\text{times}}\dots\underbrace{0\dots0}_{\mu\lambda}$. Let i' and i'' be respectively the primary pivot and the secondary pivot of $\mathbf{v}$. We can consider a simpler situation, that is, $i' = 1$ and $s\mu + 2 \leq i'' \leq s\mu + \mu$. In fact, if $i' \neq 1$, then $(0^{(\lambda+1)\mu}\Delta)(0^{\lambda}\Delta)^{s-1} \preccurlyeq \mathbf{v}$ and we have two cases:

i) if $s \geq \lambda + 3$ then $s - 1 \geq \lambda + 2$ and the bound would be satisfied since it holds:

$$d \geq (\lambda+1)\mu + \mu + s - 1 - \lambda - 2 \geq (\lambda+1)\mu + s - \lambda - 1 \geq \lambda\mu + \mu + s - \lambda - 1;$$

ii) if $s = \lambda + 1, \lambda + 2$ then $1 \geq s - (\lambda + 1)$ and so from the BCH bound we have:

$$d \geq \lambda\mu + \mu + 1 \geq (\lambda+1)\mu + s - \lambda - 1 = (\lambda+1)\mu + s - (\lambda+1).$$

As regards $s\mu + 2 \leq i'' \leq s\mu + \mu$, if it does not hold we have $(0^{\mu\lambda}\Delta)(0^{\mu-1}\Delta)^{s+1} \preccurlyeq \mathbf{v}$ and

$$\begin{aligned} d &\geq \mu\lambda + \mu + s + 1 - \lambda - 1 \\ &\geq \mu\lambda + \mu + s - \lambda - 1. \end{aligned}$$

In a similar way to the proof of Proposition 6.0.5 we are going to choose $\lambda\mu+\mu+s$ rows of $M(\mathbf{v})$. We collect the first $(\lambda\mu+\mu)$ rows of $M(\mathbf{v})$ in a matrix T_1, noting that they are the row with the primary pivot in first position and its shifts up to the $(\lambda\mu+\mu-1)-$th shift (included), so:

$$T_1=\left(\begin{array}{ccccccccccccccccccccccccc}
\Delta^+ & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & 0 & \dots & \dots & \dots & 0 & \rightarrow & 1 \\
0 & \Delta^+ & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & 0 & \dots & \dots & \dots & & \\
0 & 0 & \Delta^+ & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & 0 & \dots & \dots & & \\
0 & \dots & 0 & \Delta^+ & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & \dots & \dots & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
\Delta & \Delta & 0 & \dots & 0 & \Delta^+ & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
\Delta & \dots & \Delta & 0 & \dots & 0 & \Delta^+ & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \rightarrow & \lambda\mu+\mu \\
 & & & \downarrow & & & \downarrow & & & & & & & & & & & & & & & & & \\
 & & & \mu & & & \lambda\mu+\mu & & & & & & & & & & & & & & & & &
\end{array}\right).$$

In T_1 we note that for any row h and any column $\mu \le j \le (s-1)\mu$ we have:

$$T_1[h,j]=\Delta \implies T_1[h,j+\mu]=\Delta \tag{6.11}$$

We recall that T_1 has full rank as proved in Theorem 3.4.35.

Lemma 6.1.9. *The singleton procedure is successful for* T_1 *and thus* $\mathrm{rk}(T_1)=\lambda\mu+\mu$.

Proof. See Theorem 3.4.35. □

Then any matrix containing T_1 has rank at least $\lambda\mu+\mu$, and we obtain (6.8). If $(\mu,n)\le\mu-1$ (which it holds if and only if $\mu\nmid n$, since $\mu\le n$), then we consider another matrix, T_2, in which we collect s rows of $M(\mathbf{v})$: the $((n-i''+k\mu)_n+1)-$th rows with $k=1,\dots,s$, which are the rows with the secondary pivot in position $k\mu$.

$$T_2=\left(\begin{array}{cccccccccccccccccccccccccccccc}
\dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & \rightarrow & 1 \\
\dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & \rightarrow & 2 \\
\dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & \rightarrow & 3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & & & & \\
\dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & \dots & \dots & \dots & \dots & 0 & \rightarrow & s-1 \\
\dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^+ & \dots & 0 & \rightarrow & s \\
 & & & & \downarrow & & & & \downarrow & & & & \downarrow & & & & & & & & & \downarrow & & & & \downarrow & & & & \\
 & & & & \mu & & & & 2\mu & & & & 3\mu & & & & & & & & & (s-1)\mu & & & & \mu & & & &
\end{array}\right)$$

Note that there may be some rows in common between T_1 and T_2.

Lemma 6.1.10. *The singleton procedure is successful for* T_2 *and thus* $\mathrm{rk}(T_2)=s$. *Moreover, the* h*-th row of* T_2 *is the row corresponding to the singleton* $T_2[h\mu]$ *for* $1\le h\le s$.

Proof. The rows in T_2 correspond to the rows of matrix T_3 in the proof of Proposition 6.0.5, but a shift and a permutation, so it is enough to apply Lemma 6.1.3 and Lemma 3.2.9-3.2.10. □

Our aim is to put together the rows of T_1 and T_2, obtaining a matrix T, and identifying a submatrix $\widetilde{T}$ of T, where we apply the singleton procedure.

$$T=\left(\begin{array}{cccccccccccccccccccccccc}
\Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & \dots & \rightarrow & 1\\
0 & \Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & 0 & 0 & 0 & \Delta & \dots & \dots & \dots & \dots & & \\
0 & 0 & \Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & 0 & 0 & 0 & \Delta & \dots & \dots & \dots & & \\
0 & 0 & 0 & \Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & 0 & 0 & 0 & \Delta & \dots & \dots & & \\
0 & \dots & \dots & 0 & \Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & 0 & 0 & 0 & \Delta & \dots & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
\Delta & 0 & \dots & \dots & \dots & \dots & 0 & \Delta & \dots & \Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & & \\
\Delta & \Delta & 0 & \dots & \dots & \dots & \dots & 0 & \Delta & \dots & \Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & & \\
\Delta & \dots & \Delta & 0 & \dots & \dots & \dots & \dots & 0 & \Delta & \dots & \Delta^{+} & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \rightarrow & \lambda\mu{+}\mu\\
\hline
0 & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \rightarrow & 1\\
0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
\dots & \dots & 0 & \dots & 0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \\
0 & \Delta & 0 & \dots & 0 & \Delta & \dots & \dots & \dots & \dots & 0 & \dots & 0 & \Delta & \dots & \Delta^{+} & \dots & \dots & \dots & \dots & \dots & \dots & \rightarrow & s\\
 & & & \downarrow & & & & \downarrow & & & & \downarrow & & & & \downarrow & & & & & & & & \\
 & & & \mu & & & & \lambda\mu & & & & \dots & & & & s\mu & & & & & & & &
\end{array}\right)$$

In order to do that, we use the singletons of the matrix T_2, removing, if necessary, some rows of T_1. Let $k = 1, \dots, s$ and $B_{k\mu}$ be the set of the rows of T_1 to discard so that $T(k\mu)$ become a singleton. In other words, $B_{k\mu} = \{\, h \mid T_1[h, k\mu] = \Delta \,\}$. To determine the maximal number of the discarded rows of T_1, we have to estimate the size of $\mathbf{B} = \cup_{k=1}^{s} B_{k\mu}$. Thanks to (6.11), if $k' \leq k$ then $B_{k'\mu} \subseteq B_{k\mu}$, so $\mathbf{B} = B_{s\mu}$ and it is enough to estimate

$$\begin{aligned}
\eta &= |\{\, h \mid T_1[h, s\mu] = \Delta, 1 \leq h \leq \lambda\mu + \mu \,\}| \\
&= |\{\, h \mid T_1[1, s\mu - h] = \Delta, 0 \leq h \leq \lambda\mu + \mu - 1 \,\}|. \\
&= |\{\, h \mid \mathbf{v}[s\mu - h] = \Delta, 0 \leq h \leq \lambda\mu + \mu - 1 \,\}|.
\end{aligned}$$

Since $s \geq \lambda + 1$, starting from $\mathbf{v}[s\mu]$ and moving to the left of $(\lambda\mu + \mu)$ positions, we meet exactly $\lambda + 1$ blocks $(0^{\mu-1}\Delta)$, each contributing to η by at most 1, so $\eta \leq \lambda + 1$.

Remark 6.1.11. Note that for the computation of η we did not need to use Lemma 6.1.6, since this time we know exactly where the secondary pivot is, thus the determination of η is easier.

In conclusion, we have just proved that discarding at most $\lambda + 1$ rows of T, we obtain a submatrix $\widetilde{T}$ of T for which the singleton procedure is successful and we conclude:

$$\mathrm{rk}(T) \geq \mathrm{rk}(\widetilde{T}) = \lambda\mu + \mu + s - \lambda - 1. \quad \square$$

$\square$

Example 6.1.12. Let C be a cyclic code of length $n = 27$, with defining set S_C satisfying the assumptions of Proposition 6.0.9 with parameters $\mu = 4$, $\lambda = 2$, $s = 5$. We want to prove that by Proposition 6.0.9 the distance of the code C is at least $d \geq 4 \cdot 2 + 4 + 4 - 2 - 1 = 13$. Let $\mathbf{v} \in \mathcal{A}(R(n, S_C))$, then we can suppose $\mathbf{v}[1] = \Delta^{+}$

and $i'' = 18$ or $i'' = 19$, otherwise the bound is trivially satisfied.
Case 1: $i'' = 18$, $\mathbf{v} = \Delta^{+}000\Delta000\Delta000\Delta000\Delta\Delta^{+}\Delta00000000$.

$\Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow$ 5-th s-deletion

$0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow$ 6-th s-deletion

$0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow$ 7-th s-deletion

~~$0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0$~~ $\rightarrow$ **DISCARDED**

$0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0 \rightarrow$ 8-th s-deletion

$0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0 \rightarrow$ 9-th s-deletion

$0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0 \rightarrow$ 10-th s-deletion

~~$0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0$~~ $\rightarrow$ **DISCARDED**

$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta \rightarrow$ 11-th s-deletion

$\Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta \rightarrow$ 12-th s-deletion

$\Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta \rightarrow$ 13-th s-deletion

~~$\Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0$~~ $\rightarrow$ **DISCARDED**

$0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0 \rightarrow$ 1-st s-deletion

$0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0 \rightarrow$ 2-nd s-deletion

$0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0 \rightarrow$ 3-rd s-deletion

$0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta^{+}\ \Delta\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0 \rightarrow$ 4-th s-deletion

Case 2: $i'' = 19$, $\mathbf{v} = \Delta^{+}000\Delta000\Delta000\Delta000\Delta\Delta\Delta^{+}00000000$.

$\Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow$ 5-th s-deletion

$0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow$ 6-th s-deletion

$0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0 \rightarrow$ 7-th s-deletion

~~$0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0$~~ $\rightarrow$ **DISCARDED**

$0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0 \rightarrow$ 8-th s-deletion

$0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0 \rightarrow$ 9-th s-deletion

$0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0 \rightarrow$ 10-th s-deletion

~~$0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0$~~ $\rightarrow$ **DISCARDED**

$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+} \rightarrow$ 11-th s-deletion

$\Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta \rightarrow$ 12-th s-deletion

$\Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta \rightarrow$ 13-th s-deletion

~~$\Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0$~~ $\rightarrow$ **DISCARDED**

$0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0 \rightarrow$ 1-st s-deletion

$0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0 \rightarrow$ 2-nd s-deletion

$0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0\ 0\ \Delta\ 0\ 0 \rightarrow$ 3-rd s-deletion

$0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ 0\ 0\ 0\ \Delta\ \Delta\ \Delta^{+}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \Delta^{+}\ 0\ 0 \rightarrow$ 4-th s-deletion

We summarize the results of Proposition 6.0.5 and Proposition 6.0.9 in one statement, called bound C.

Theorem 6.1.13 (Bound C)**.** *Let C be a $[n,k,d]$ cyclic code with defining set S_C. Suppose that there are $\ell,\ m,\ r,\ s,\ \rho \in \mathbb{N}$, $1 \leq m \leq \ell$, $s \geq 1$, $\rho \geq 1$ such that*

$$((0)^{\ell}(\Delta)^{r})((0)^{m}(\Delta)^{r})^{s} \preccurlyeq R(n,S_C)^{\rho} \qquad \textit{or} \qquad ((\Delta)^{r}(0)^{m})^{s}((\Delta)^{r}(0)^{\ell}) \preccurlyeq R(n,S_C)^{\rho}.$$

Then:

- *if $(m+r,n) \leq m$:*

$$d \geq \ell + 1 + s - r\left\lfloor \frac{\ell}{m+r} \right\rfloor - \max\{\, (\ell)_{m+r} - \lambda, 0 \,\};$$

- *otherwise:*

$$d \geq \ell + 1.$$

In the particular case that, $\ell = \lambda\mu$, $m = \mu - 1$, $s \geq \lambda + 1$ and $r = 1$ for some μ and λ, we also have:

- $d \geq \mu\lambda + \mu + s - \lambda - 1$, *if* $\mu \nmid n$
- $d \geq \mu\lambda + \mu$, *otherwise.*

As explained in Remark 6.0.7 and in Remark 6.0.11 bound `C` is both a generalization of the HT bound and the BS bound (except when $\mu|n$) and so it is sharper and tighter. Our bound and the Roos bound are independent, in fact one is a strict root bound, while the other it is not. As a consequence, for some codes our bound is sharper and tighter than Roos's but for other codes it is the opposite. From the computed codes, it appears that bound `C` is tighter than the Roos bound in the majority of cases. and so, Bound `C` is the first polynomial-time bound outperforming the Roos bound on a significant sample of codes.

Remark 6.1.14. Also the BS bound and the Roos bound are independent, and indeed the BS bound for some codes beats the Roos bound. However, in the majority of computed cases the Roos bound is better, as reported in [BS06] and checked by us.

As regards computational costs, bound `C` requires at most:

- n operations for i_0
- n operations for ℓ,
- n operations for m,
- n operations for r,
- n operations for s

and so it costs $O(n^5)$ which is slightly more than the Roos bound which needs $O(n^4)$, in fact the latter requires at most:

- n operations for i_0,
- n operations for m,
- n operations for r,
- n operations for s

while the other bounds cost less: BCH $\rightarrow O(n^2)$, HT $\rightarrow O(n^3)$, bound BS $\rightarrow O(n^{2.5})$. We tested all cyclic codes in the following range: on $\mathbb{F}_2$ with $15 \leq n \leq 125$, on $\mathbb{F}_3$ with $8 \leq n \leq 79$ and $82 \leq n \leq 89$, on $\mathbb{F}_5$ with $8 \leq n \leq 61$, on $\mathbb{F}_7$ with $8 \leq n \leq 47$. We have chosen the largest ranges that we could compute in a reasonable time. In Table 6.2- 6.3- 6.4- 6.5- 6.6- 6.7 we give in detail the results obtained for each characteristic. We write BCH for the BCH bound, HT for the HT bound, BS for the BS bound,RS for the Roos bound and BC for the bound `C`.

Since all the bounds that we consider are sharper than the BCH bound, clearly they are tight for all cyclic codes in which the BCH bound is already tight. Thus, it is interesting to consider the only cases when the HT, BS, Roos and `C` bounds are tight and the BCH bound is not.

The following table summarizes our findings and is composed of two different parts. In the first part we report: in the first row the number of checked codes, in the second row the number of these for which the BCH bound is tight. In the second part of the table, each row corresponds to a specific bound. For each row we report the number of codes for which the bound is tight and the BCH bound is not.

Table 6.1: Bound tightness

	$\mathbb{F}_2$	$\mathbb{F}_3$	$\mathbb{F}_5$	$\mathbb{F}_7$	total
number of codes	70488	93960	1163176	106804	1434428
BCH	59296	77584	1011957	93108	1241945
HT	661	1042	12058	2603	16364
BS	233	831	11436	2413	14913
ROOS	**1178**	1793	17673	2987	23631
bound `C`	886	**1811**	**20147**	**4155**	**26999**

n	N_{codes}	BCH	HT	BS	RS	BC
15	32	30	32	30	32	32
17	8	5	8	5	8	8
19	4	4	4	4	4	4
21	64	52	54	52	58	54
23	8	4	4	4	4	4
25	8	8	8	8	8	8
27	16	16	16	16	16	16
29	4	4	4	4	4	4
31	128	46	96	46	96	96
33	32	21	26	21	26	26
35	64	40	42	40	48	44
37	4	4	4	4	4	4
39	32	18	20	18	20	20
41	8	4	4	4	4	4
43	16	6	10	6	11	10
45	256	187	222	189	228	224
47	8	4	4	4	4	4
49	32	32	32	32	32	32
51	256	90	146	98	146	150
53	4	4	4	4	4	4
55	32	16	20	16	20	20
57	32	20	24	20	24	24
59	4	4	4	4	4	4
61	4	4	4	4	4	4
63	8192	2238	4210	2401	4346	4280
65	128	36	74	36	78	74
67	4	4	4	4	4	4
69	64	22	24	22	24	24

Table 6.2: Tightness $\mathbb{F}_2$, $15 \leq n \leq 69$

n	N_{codes}	BCH	HT	BS	RS	BC
71	8	4	4	4	4	4
73	512	37	104	39	117	106
75	256	220	252	220	254	252
77	64	42	44	42	44	44
79	8	4	4	4	4	4
81	32	32	32	32	32	32
83	4	4	4	4	4	4
85	4096	547	1124	571	1141	1132
87	32	18	20	18	20	20
89	512	20	56	20	56	56
91	1024	277	435	277	436	435
93	16384	1388	3268	1424	3360	3286
95	32	18	20	18	20	20
97	8	4	4	4	4	4
99	256	105	166	106	171	166
101	4	4	4	4	4	4
103	8	4	4	4	4	4
105	32768	7939	11446	8420	12325	11796
107	4	4	4	4	4	4
109	16	4	4	4	4	4
111	32	18	20	22	20	24
113	32	4	4	4	4	4
115	64	24	26	24	26	26
117	4096	637	1075	714	1110	1099
119	512	170	212	170	213	212
121	8	8	8	8	8	8
123	256	52	62	60	62	66
125	16	16	16	16	16	16

Table 6.3: Tightness $\mathbb{F}_2$, $71 \leq n \leq 125$

n	N_{codes}	BCH	HT	BS	RS	BC
8	32	30	32	30	32	32
10	16	16	16	16	16	16
11	8	4	4	4	4	4
13	32	19	26	19	27	26
14	16	16	16	16	16	16
16	128	112	118	112	120	118
17	4	4	4	4	4	4
19	4	4	4	4	4	4
20	128	90	102	100	104	110
22	64	24	24	32	24	32
23	8	4	4	4	4	4
25	8	8	8	8	8	8
26	1024	321	514	377	545	546
28	128	94	116	96	120	120
29	4	4	4	4	4	4
31	4	4	4	4	4	4
32	512	410	464	414	472	464
34	16	16	16	16	16	16
35	32	16	18	16	20	18
37	8	4	4	4	4	4
38	16	16	16	16	16	16
40	8192	3170	4344	3570	4478	4614
41	64	9	29	9	30	29
43	4	4	4	4	4	4
44	512	208	216	236	218	244
46	64	24	24	24	24	24
47	8	4	4	4	4	4
49	8	8	8	8	8	8
50	64	64	64	64	64	64
52	32768	7157	11452	8281	12339	12150

Table 6.4: Tightness $\mathbb{F}_3$, $8 \leq n \leq 52$

n	N_{codes}	BCH	HT	BS	RS	BC
53	4	4	4	4	4	4
55	64	20	22	20	24	22
56	8192	3168	4368	3414	4440	4466
58	16	16	16	16	16	16
59	8	4	4	4	4	4
61	128	5	10	5	11	10
62	16	16	16	16	16	16
64	2048	1640	1866	1652	1916	1870
65	1024	211	324	211	351	324
67	16	4	4	4	4	4
68	128	76	88	76	88	88
70	1024	422	454	450	464	490
71	8	4	4	4	4	4
73	128	5	10	5	10	10
74	64	28	32	28	32	32
76	128	92	112	92	112	112
77	64	20	22	20	22	22
79	4	4	4	4	4	4
82	4096	303	799	303	798	799
83	8	4	4	4	4	4
85	128	30	36	30	40	36
86	16	16	16	16	16	16
88	32768	8952	11484	9928	11866	12042
89	4	4	4	4	4	4
92	512	196	204	196	204	204
94	64	24	24	24	24	24
95	32	18	20	18	20	20
97	8	4	4	4	4	4
98	64	64	64	64	64	64

Table 6.5: Tightness $\mathbb{F}_3$, $53 \leq n \leq 98$, $n \neq 80$, $n \neq 91$

n	N_{codes}	BCH	HT	BS	RS	BC
8	32	26	32	26	32	32
9	32	32	32	32	32	32
10	16	16	16	16	16	16
11	4	4	4	4	4	4
12	512	458	488	482	488	500
13	4	4	4	4	4	4
15	64	58	64	58	64	64
16	512	218	326	250	336	342
17	4	4	4	4	4	4
18	1024	952	988	988	988	1012
19	128	14	28	18	28	28
20	128	82	94	88	96	98
22	16	16	16	16	16	16
23	4	4	4	4	4	4
24	32768	15416	21794	17762	21836	22976
25	128	28	72	29	74	72
26	16	16	16	16	16	16
27	128	128	128	128	128	128
29	32	4	4	4	4	4
30	4096	2614	2890	3046	2914	3323
31	8	4	4	4	4	4
32	8192	2258	3518	2480	3652	3638
33	64	58	64	58	64	64
34	16	16	16	16	16	16
36	32768	25346	27860	27890	28124	29204
37	32	4	4	4	4	4
38	16384	762	1610	946	1746	1730
39	64	58	64	58	64	64
40	8192	2664	3598	2952	3696	3746
41	4	4	4	4	4	4
43	256	4	6	4	6	6
44	128	84	96	84	98	96
45	1024	763	850	763	856	850
46	16	16	16	16	16	16
47	8	4	4	4	4	4

Table 6.6: Tightness $\mathbb{F}_7$, $8 \leq n \leq 47$

n	N_{codes}	BCH	HT	BS	RS	BC
8	64	60	64	60	64	64
9	8	8	8	8	8	8
11	8	4	4	4	4	4
12	256	204	220	228	224	236
13	16	7	14	8	14	14
14	16	16	16	16	16	16
16	256	240	252	240	256	252
17	4	4	4	4	4	4
18	64	64	64	64	64	64
19	8	4	4	4	4	4
21	32	20	24	20	24	24
22	64	24	24	32	24	32
23	4	4	4	4	4	4
24	16384	7264	10280	8276	10560	10720
26	256	81	156	92	156	160
27	16	16	16	16	16	16
28	256	208	224	208	240	224
29	8	4	4	4	4	4
31	2048	69	225	73	242	229
32	1024	972	1008	972	1024	1008
33	64	22	24	22	24	24
34	16	16	16	16	16	16
36	4096	2308	2936	3084	3196	3280
37	4	4	4	4	4	4
38	64	24	24	24	24	24
39	2048	244	423	267	429	427
41	8	4	4	4	4	4
42	1024	504	702	530	706	702
43	4	4	4	4	4	4
44	4096	1484	1696	1692	1716	1840
46	16	16	16	16	16	16
47	4	4	4	4	4	4
48	1048576	400240	561252	445932	572536	579044
49	8	8	8	8	8	8
51	32	18	20	18	20	20
52	65536	13265	18552	15032	18676	19160
53	4	4	4	4	4	4
54	256	256	256	256	256	256
56	16384	6780	8396	7428	8824	8788
57	64	22	24	22	24	24
58	64	28	32	28	32	32
59	8	4	4	4	4	4
61	8	4	4	4	4	4

Table 6.7: Tightness $\mathbb{F}_5$, $8 \leq n \leq 61$

Proving some root bounds via Newton's identities

In Chapter 5 we presented two different methods exploiting Gröbner bases in order to find the minimum distance of a cyclic code C. These methods are based on solving two different kinds of polynomial systems, indexed by w, $\hat{J}_C(w)$ and $\mathcal{S}_C(w)$, which establish the existence of words of weight w in C. In both systems, information on $\mathbb{F}_q$, the ground field of C, is used to bound the distance of the code. To be more precise, the field $\mathbb{F}_q$ appears in the equations $y_i^{q-1} - 1 = 0$, $1 \leq i \leq w$, for $\hat{J}_C(w)$ and in the equations $S_{qi \mod n} = S_i^q$, $0 \leq i \leq n-1$, for $\mathcal{S}_C(w)$. However, the root bounds introduced in Section 3.2 allow to estimate the distance of C without any knowledge on the ground field, provided the length and the defining set are known. In fact, we will see in this chapter's proofs that the equations depending on $\mathbb{F}_q$ are unnecessary. We will focus on the approach with Newton's identities, showing how the strict root bounds proposed in Section 3.4 can be proved removing the constraints $S_{qi \mod n} = S_i^q$, $0 \leq i \leq n-1$.

The main results that we claim in this chapter are:

- a polynomial proof of the HT bound in the more general version by Roos; this is the first proof of that bound by polynomials, although a polynomial proof for the special case $(m+r, n) = 1$ was given in [HT72]
- a polynomial proof of BS bound
- a polynomial proof for the Boston bound I,II, III, IV.

7.1 A polynomial interpretation of known strict root bounds

In Section 3.4 we proposed an alternative formulation of many well-known bounds that we actually proved to be strict root bounds. In this section we give another statement for each of these bounds, based on the definition of DFT. For this reason, sometimes we call them the `spectral definition` of the bounds. We adopt the same notation of Section 5.3: for any word c of a cyclic code $C \in \mathcal{C}_{q,n}$, of weight w, we indicate with X_i, $1 \leq i \leq w$ the locations of c and with Y_i, $1 \leq i \leq w$, the values of c. We also write $S_j = \sum_{i=1}^{w} Y_i X_i^j$ and note $S_{j+n} = S_j$. Note also that $S_j = 0$ for n consecutive S_j's if and only if $c = 0$.

7.1.1 A polynomial interpretation of the BCH bound

We now provide the spectral definition of the BCH bound, and prove it using Newton's identities. This solves the Problem (55) in [MS77] as was already proved by Chien in [Chi72].

Theorem 7.1.1. *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$. If for all $c \in C$ there are i, $\ell \in \{0,\dots,n-1\}$ such that*

$$S_{i+k} = 0, \quad 0 \le k \le \ell - 1$$

Then

$$d \ge \ell + 1.$$

Proof. Let c be any non-zero word of C of weight $1 \le w \le \ell$. By hypothesis there exists i such that $S_i = \dots = S_{i+\ell-1} = 0$. We prove by induction that $S_{i+\ell+k} = 0$ for $k \ge 0$.
Let us consider $k = 0$; from the generalized Newton identities (5.3) we have:

$$\forall\, j \ge 0, \quad S_{j+w} + \sigma_1 S_{j+w-1} + \dots + \sigma_w S_j = 0. \tag{7.1}$$

In particular, for $j = i + \ell - w$, we obtain

$$0 = S_{i+\ell} + \sigma_1 S_{i+\ell-1} + \dots + \sigma_w S_{i+\ell-w}\,.$$

The right hand side of previous equation reduces to $S_{i+\ell}$, since $S_i = \dots = S_{i+\ell} = 0$ by hypothesis and so $S_{i+\ell} = 0$. We suppose by inductive hypothesis that $S_{i+\ell+k} = 0$ for $0 \le k \le \bar{k} - 1$ and we prove $S_{i+\ell+\bar{k}} = 0$. Substituting $j = i + \ell + \bar{k} - w$ in (7.1), we obtain:

$$0 = S_{i+\ell+\bar{k}} + \sigma_1 S_{i+\ell+\bar{k}-1} + \dots + \sigma_w S_{i+\ell+\bar{k}-w} = S_{i+\ell+\bar{k}}.$$

From $S_i = S_{i+n}$ we have that for $k \ge 0$, $S_{i+\ell+k} = S_{(i+\ell+k)_n}$, thus $S_0 = \dots = S_{n-1} = 0$ and the claim is proved. □

Remark 7.1.2. In the proof of Theorem 7.1.1, we have shown that for any $1 \le w \le \ell$ the unique word of C which satisfies

$$\begin{cases} S_{w+k} + S_{w+k-1}\sigma_1 + \dots + S_k \sigma_w = 0, & 0 \le k \le n-1 \\ S_{k+n} = S_k, & 0 \le k \le n-1 \\ S_k = 0, & \forall\, k \in \{i, i+1, \dots, i+\ell-1\} \end{cases} \tag{7.2}$$

is the zero codeword. In particular, the equations in (7.2) are the same of $\mathcal{S}_C(w)$, execpt for the equations regarding the field $\mathbb{F}_q$, which are the unnecessary in the proof of the BCH bound

7.1.2 A polynomial interpretation of the HT bound

Here we propose the spectral formulation of the Hartmann-Tzeng bound ([HT72]), as generalized by C. Roos in [Roo82].

Theorem 7.1.3 (Hartmann-Tzeng bound, [Roo82])**.** *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$. Suppose that for all $c \in C$ there exist $\ell, m, s, r \in \mathbb{N}$ s.t. $m \geq 1$, $s \geq 1$, $(m+r, n) \leq m$ for which*

$$S_{\ell+i+j(m+r)} = 0, \quad 0 \leq i \leq m-1,\ 0 \leq j \leq s-1\,. \tag{7.3}$$

Then

$$d \geq m+s.$$

Proof. We can suppose by the BCH bound that $d \geq m+1$. Let us consider a word c of weight $\mathrm{w}(c) = w$, $m+1 \leq w \leq m+s-1$. We consider two polynomials:

$$p(z) = \prod_{i=1}^{m}(z - X_i) = \sum_{i=0}^{m} p_{m-i} z^i, \qquad \text{where } p_0 = 1$$

$$q(z) = \prod_{i=m+1}^{w}(z^{m+r} - X_i^{m+r}) = \sum_{j=0}^{w-m} q_{w-m-j} z^{j(m+r)}, \qquad \text{where } q_0 = 1.$$

Let $\sigma(z)$ be the product of $p(z)$ and $q(z)$:

$$\sigma(z) = \sum_{i=0}^{m}\sum_{j=0}^{w-m} p_{m-i} q_{w-m-j} z^{i+j(m+r)}.$$

Although $\sigma(z)$ is not the plain locator polynomial of c, it is a multiple of it. Since $\sigma(X_t) = 0$ for any $1 \leq t \leq w$, we have:

$$\begin{aligned} 0 = \sum_{t=1}^{w} X_t^k Y_t \sigma(X_t) &= \sum_{t=1}^{w}\left(\sum_{i=0}^{m}\sum_{j=0}^{w-m} p_{m-i} q_{w-m-j} X_t^{i+j(m+r)+k} Y_t\right) \\ &= \sum_{i=0}^{m}\sum_{j=0}^{w-m} p_{m-i} q_{w-m-j}\left(\sum_{t=1}^{w} X_t^{i+j(m+r)+k} Y_t\right) \\ &= \sum_{i=0}^{m}\sum_{j=0}^{w-m} p_{m-i} q_{w-m-j} S_{i+j(m+r)+k}. \end{aligned} \tag{7.4}$$

We claim that for any $k \geq \ell$:

$$\sum_{j=0}^{w-m} q_{(w-m)-j} S_{m+j(m+r)+k} = 0. \tag{7.5}$$

We assume for the moment that (7.5) holds and we postpone the proof. By Lemma 3.4.17, we can suppose without loss of generality that there exists $m' \in \{ 0, \ldots, m-1 \}$ such that $S_{s(m+r)+\ell+m'} \neq 0$, otherwise we could increase the distance of one and the proof proceeds similarly. Let us substitute $k = m' - m + \ell + (s - (w-m))(m+r)$ in (7.5), noting that $k \geq -m + \ell + (s-(s-1))m \geq \ell$:

$$\begin{aligned} 0 &= \sum_{j=0}^{w-m} q_{w-m-j} S_{m+(m'-m+\ell+(s-(w-m))(m+r))+j(m+r)} \\ &= \sum_{j=0}^{w-m} q_{w-m-j} S_{m'+\ell+(s-(w-m)+j)(m+r))}. \end{aligned} \tag{7.6}$$

Let $j' = (s + j - (w-m))$ in (7.6), then:

$$\begin{aligned} 0 &= \sum_{j'=s-(w-m)}^{s} q_{s-j'} S_{\ell+j'(m+r)+m'} \\ &= \sum_{j'=s-(w-m)}^{s-1} q_{s-j'} S_{\ell+j'(m+r)+m'} + S_{\ell+s(m+r)+m'} = S_{\ell+s(m+r)+m'}. \end{aligned}$$

Since for $1 \leq s-(w-m) \leq j' \leq s-1$ and $0 \leq m' \leq m-1$, we have $S_{\ell+j'(m+r)+m'} = 0$ by hypothesis. But $S_{\ell+s(m+r)+m'} \neq 0$ and we get the contradiction.

Proof of (7.5). We show (7.5) by induction. If $k = \ell$, substituting in (7.4) we obtain:

$$\begin{aligned} 0 &= \sum_{i=0}^{m} \sum_{j=0}^{w-m} p_{m-i} q_{w-m-j} S_{i+j(m+r)+\ell} \\ &= \sum_{i=0}^{m-1} \sum_{j=0}^{w-m} p_{m-i} q_{w-m-j} S_{i+j(m+r)+\ell} + \sum_{j=0}^{w-m} p_0 q_{w-m-j} S_{m+j(m+r)+\ell} \end{aligned} \tag{7.7}$$

Noting that, by assumption, $0 \leq j \leq w-m \leq s-1$, from (7.3) and (7.7) we have:

$$0 = \sum_{j=0}^{w-m} p_0 q_{w-m-j} S_{m+j(m+r)+\ell} = \sum_{j=0}^{w-m} q_{w-m-j} S_{m+j(m+r)+\ell}.$$

Let us suppose that (7.5) holds for any t, $\ell \leq t \leq k'$ and we prove it for $k'+1$. Substituting $k = k'+1$ in (7.4), we have:

$$\begin{aligned} 0 &= \sum_{i=0}^{m} \sum_{j=0}^{w-m} p_{m-i} q_{w-m-j} S_{i+j(m+r)+k'+1} \\ &= \sum_{i=0}^{m-1} \sum_{j=0}^{w-m} p_{m-i} q_{w-m-j} S_{i+j(m+r)+k'+1} + \sum_{j=0}^{w-m} q_{w-m-j} S_{m+j(m+r)+k'+1}. \end{aligned}$$

To conclude we have to prove: $\sum_{i=0}^{m-1}\sum_{j=0}^{w-m} p_{m-i}q_{w-m-j}S_{i+j(m+r)+k'+1} = 0$. Let $b = k' - \ell$ and $i' = i+1$, then:

$$\sum_{i=0}^{m-1}\sum_{j=0}^{w-m} p_{m-i}q_{w-m-j}S_{i+j(m+r)+k'+1} = \sum_{i'=1}^{m}\sum_{j=0}^{w-m} p_{m-i'+1}q_{w-m-j}S_{\ell+i'+j(m+r)+b} =$$
$$= \sum_{i'=1}^{(m-1)-b}\sum_{j=0}^{w-m} p_{m-i'+1}q_{w-m-j}S_{\ell+i'+j(m+r)+b} + \sum_{i'=m-b}^{m}\sum_{j=0}^{w-m} p_{m-i'+1}q_{w-m-j}S_{\ell+i'+j(m+r)+b}. \tag{7.8}$$

But for $1 \le i' \le (m-1)-b$ we have $0 \le i'+b \le m-1$ and by (7.3) the first term is zero, so (7.8) becomes: $\sum_{i'=m-b}^{m}\sum_{j=0}^{w-m} p_{m-i'+1}q_{w-m-j}S_{\ell+i'+j(m+r)+b}$.

Let $(\ell+i'+b)-m-1 = h$, then:

$$\sum_{i'=m-b}^{m} p_{m-i'+1}\sum_{j=0}^{w-m} q_{w-m-j}S_{\ell+i'+j(m+r)+b} = \sum_{h=\ell-1}^{\ell+b-1} p_{b+\ell-h}\sum_{j=0}^{w-m} q_{w-m-j}S_{m+h+j(m+r)+1} =$$
$$= \sum_{h=\ell-1}^{k'-1} p_{k'-h}\sum_{j=0}^{w-m} q_{w-m-j}S_{m+h+j(m+r)+1}.$$

Setting $t = h+1$ we get

$$\sum_{h=\ell-1}^{k'-1} p_{k'-h}\sum_{j=0}^{w-m} q_{w-m-j}S_{m+h+j(m+r)+1} = \sum_{t=\ell}^{k'} p_{k'-t+1}\sum_{j=0}^{w-m} q_{w-m-j}S_{m+j(m+r)+t} = 0,$$

by inductive hypothesis. So (7.5) holds. □

Remark 7.1.4. We note that, differently from Theorem 7.1.1, in the proof of Theorem 7.1.3 we do not use the generalized Newton identities, presented in (5.3), but we propose a new kind of identities: $0 = \sum_{i=0}^{m}\sum_{j=0}^{w-m} p_{m-i}q_{w-m-j}S_{i+j(m+r)+k}$, for suitable coefficients p_i, q_j. Observe also that we still not use any condition involving the field $\mathbb{F}_q$.

We provide an example to explain the technical details of the proof of Theorem 7.1.3.

Example 7.1.5. Let C be an $[n,k,d]$ code over (any) $\mathbb{F}_q$ of length $n \ge 18$ and $5 \nmid n$, which contains in its defining set $J = \{1,2,3,6,7,8,11,12,13\}$. From the HT bound, setting $m=3$, $r=2$, $k=1$, we get $d \ge 6$. We can suppose that one between S_{16}, S_{17} and S_{18} is different from zero. Let $c \in C$ be a word of weight $\mathrm{w}(c) = 5$, with locations X_1, X_2, X_3, X_4, X_5, values Y_1, Y_2, Y_3, Y_4, Y_5 and $\mathrm{DFT}(c) = (S_0,\dots,S_{n-1})$. We have that $S_j = 0$ for any $j \in J$. Let us consider two polynomials:

$$\begin{aligned} p(z) &= (z - X_1)(z - X_2)(z - X_3) & q(z) &= (z^5 - X_4^5)(z^5 - X_5^5) \\ &= z^3 + p_1 z^2 + p_2 z + p_3 & &= z^{10} + q_1 z^5 + q_2 \\ &= p_0 z^3 + p_1 z^2 + p_2 z + p_3 & &= q_0 z^{10} + q_1 z^5 + q_2. \end{aligned}$$

(where $p_0 = q_0 = 1$) and their product $p(z)q(z)$:

$$z^{13} + p_1 z^{12} + p_2 z^{11} + p_3 z^{10} + q_1 z^8 + q_1 p_1 z^7 + q_1 p_2 z^6 + q_1 p_3 z^5 + q_2 z^3 + q_2 p_1 z^2 + q_2 p_2 z + q_2 p_3.$$

For any $k \geq 0$ we have:

$$\begin{aligned} 0 &= \sum_{i=1}^{5} Y_i X_i^k p(X_i) q(X_i) \\ &= \sum_{i=0}^{3} \sum_{j=0}^{2} p_{3-i} q_{2-j} S_{i+5j+k} \end{aligned} \tag{7.9}$$

Substituting in (7.9) $k = 1, \ldots, 5$, we obtain:

$$\begin{aligned} k = 1: &\quad 0 = S_{14} + q_1 S_9 + q_2 S_4 & & \\ k = 2: &\quad 0 = S_{15} + q_1 S_{10} + q_2 S_5 & & \\ k = 3: &\quad 0 = S_{16} + q_1 S_{11} + q_2 S_6 & &\Longrightarrow S_{16} = 0 \\ k = 4: &\quad 0 = S_{17} + q_1 S_{12} + q_2 S_7 & &\Longrightarrow S_{17} = 0 \\ k = 5: &\quad 0 = S_{18} + q_1 S_{13} + q_2 S_8 & &\Longrightarrow S_{18} = 0, \end{aligned}$$

which is a contradiction, since at least one between S_{16}, S_{17} or S_{18} is different from zero.

7.1.3 A polynomial interpretation of the BS bound

We provide the spectral definition of the BS bound, dividing the statement in two parts. The first part, which we call the `straight version` of the BS bound, collects the conditions a) and b) of Definition 3.4.7. The second part, which we call the `reverse version` of BS bound, collects the conditions c) and d) of Definition 3.4.7.

Theorem 7.1.6 (BS bound, straight version). *Let C be an $[n, k, d]$ code over $\mathbb{F}_q$. Suppose that there are $m, \ell \in \mathbb{N}$, $m, \ell \geq 1$ and $k \in \{0, \ldots, n-1\}$ such that $S_k = S_{k+n}$ and for all $c \in C$:*

a) $S_{k+j} = 0$, $j = 0, \ldots, m\ell - 1$,

b) $S_{k+(m+z)\ell+j} = 0$, $j = 1, \ldots, \ell - 1$, $0 \leq z \leq m$.

Then:

$$d \geq m\ell + \ell\,.$$

Proof. By the BCH bound we have $d \geq m\ell + 1$. Let us suppose that there is a non-zero word $c \in C$ of weight $m\ell + 1 \leq w \leq m\ell + \ell - 1$ and locations $X_1, \ldots, X_w$. Without loss of generality, we can suppose that $S_{k+m\ell} \neq 0$, otherwise $S_k = S_{k+1} = \cdots = S_{k+m\ell} = \cdots = S_{k+m\ell+\ell-1} = 0$ and by BCH bound we get $d \geq m\ell + \ell + 1 \geq m\ell + \ell$. For the generalized Newton identities we have that for all $j \geq 0$,

$$\sum_{i=0}^{w} S_{i+j}\sigma_{w-i} = 0, \tag{7.10}$$

where the σ_i's, $1 \leq i \leq w$, are the symmetric functions of the locations. In particular, we have $\sigma_0 = 1$, by definition, and $\sigma_w = \prod_{i=1}^{w} X_i \neq 0$.
We claim that $\sigma_{w-(m-z)\ell} = 0$ for $z = 0, \ldots, m$. Note that if our claim is true, we prove the theorem, since we get a contradiction for $z = m$, in fact $0 = \sigma_{w-(m-m)\ell} = \sigma_w \neq 0$. We proceed by induction. We start proving our claim for $z = 0$. Let us substitute $j = k$ in (7.10), we get:

$$\begin{aligned} 0 = \sum_{i=0}^{w} S_{i+k}\ \sigma_{w-i} &= \sum_{i=0}^{m\ell-1} S_{i+k}\ \sigma_{w-i} + \sum_{i=m\ell}^{w} S_{i+k}\ \sigma_{w-i} \\ &= \sum_{i=m\ell}^{w} S_{i+k}\ \sigma_{w-i} \\ &= \sum_{i=m\ell+1}^{w} S_{i+k}\sigma_{w-i} + S_{k+m\ell}\sigma_{w-m\ell}, \end{aligned}$$

setting $i' = i - m\ell$, we get

$$\begin{aligned} 0 &= \sum_{i'=1}^{w-m\ell} S_{m\ell+i'+k}\ \sigma_{w-m\ell-i} + S_{k+m\ell}\ \sigma_{w-m\ell} \\ &= S_{k+m\ell}\ \sigma_{w-m\ell}, \end{aligned}$$

because $1 \leq w - m\ell \leq \ell - q$ and by hypothesis b), $S_{m\ell+i'+k} = 0$. Thus, $\sigma_{w-m\ell} = 0$, since $S_{k+m\ell} \neq 0$. Supposing that $\sigma_{w-(m-z)\ell} = 0$ for $z < \bar{z} < m$, we prove $\sigma_{w-(m-\bar{z})\ell} = 0$. Let us substitute $j = k + \bar{z}\ell$ in (7.10):

$$\begin{aligned} 0 = \sum_{i=0}^{w} S_{i+k+\bar{z}\ell}\ \sigma_{w-i} &= \sum_{i=0}^{(m-\bar{z})\ell-1} S_{i+k+\bar{z}\ell}\ \sigma_{w-i} + \sum_{i=(m-\bar{z})\ell}^{w} S_{i+k+\bar{z}\ell}\ \sigma_{w-i} \\ &= \sum_{i=(m-\bar{z})\ell}^{w} S_{i+k+\bar{z}\ell}\ \sigma_{w-i}. \end{aligned} \tag{7.11}$$

Setting $j = i - (m - \bar{z})\ell$ in (7.11), we have:

$$\begin{aligned}
0 &= \sum_{j=0}^{w-m\ell+\bar{z}\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell} \\
&= \sum_{j=0}^{\bar{z}\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell} + \sum_{j=\bar{z}\ell+1}^{w-m\ell+\bar{z}\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell} \\
&= S_{k+m\ell}\ \sigma_{w-(m-\bar{z})\ell} + \sum_{j=1}^{\bar{z}\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell} + \sum_{j=\bar{z}\ell+1}^{w-m\ell+\bar{z}\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell}
\end{aligned} \tag{7.12}$$

Let us denote by A the summation $\sum_{j=1}^{\bar{z}\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell}$ and by B the summation $\sum_{j=\bar{z}\ell+1}^{w-m\ell+\bar{z}\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell}$. We prove that $A = 0$ and $B = 0$. We start to consider A:

$$\begin{aligned}
A &= \sum_{j=1}^{\ell} S_{j+k+m\ell}\sigma_{w-j-(m-\bar{z})\ell} + \cdots + \sum_{j=(\bar{z}-1)\ell+1}^{\bar{z}\ell} S_{j+k+m\ell}\sigma_{w-j-(m-\bar{z})\ell} \\
&= \sum_{t=0}^{\bar{z}-1} \sum_{j=t\ell+1}^{(t+1)\ell} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell} \\
&= \sum_{t=0}^{\bar{z}-1} \left(S_{(m+t+1)\ell+k}\ \sigma_{w-(m-\bar{z}+t+1)\ell} + \sum_{j=t\ell+1}^{(t+1)\ell-1} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell} \right) \\
&= \sum_{t=0}^{\bar{z}-1} S_{(m+t+1)\ell+k}\ \sigma_{w-(m-\bar{z}+t+1)\ell} + \sum_{t=0}^{\bar{z}-1} \sum_{j=t\ell+1}^{(t+1)\ell-1} S_{j+k+m\ell}\ \sigma_{w-j-(m-\bar{z})\ell}.
\end{aligned}$$

Setting $h = \bar{z} - t - 1$ in the first summation we get:

$$\sum_{t=0}^{\bar{z}-1} S_{(m+t+1)\ell+k}\ \sigma_{w-(m-\bar{z}+t+1)\ell} = \sum_{h=0}^{\bar{z}-1} S_{(m+\bar{z}-h)\ell+k}\ \sigma_{w-(m-h)\ell} = 0,$$

by inductive hypothesis ($\sigma_{w-m\ell} = \cdots = \sigma_{w-(m-(\bar{z}-1))\ell} = 0$). Similarly, also the second summation is zero, because $S_{j+m\ell+k} = 0$ for $t\ell+1 \le j \le (t+1)\ell-1$ and $0 \le t \le \bar{z}-1$, by hypothesis b). So, $A = 0$. Let us now consider B. Substituting $h = j - \bar{z}\ell$ we have:

$$B = \sum_{h=1}^{w-m\ell} S_{h+k+m\ell+\bar{z}\ell}\ \sigma_{w-h-m\ell} = 0,$$

since $S_{h+k+m\ell+\bar{z}\ell} = 0$ for $1 \le h \le (w - m\ell)$, by hypothesis. Thus (7.12) becomes $S_{k+m\ell}\ \sigma_{w-(m-\bar{z})\ell} = 0$ which implies $\sigma_{w-(m-\bar{z})\ell} = 0$ and concludes our proof. □

The proof of Theorem 7.1.6 is rather technical and requires elaborated computations, so we provide an example to clarify its details.

Example 7.1.7. Let C be an $[n,k,d]$ cyclic codes of length $n \geq 16$ over (any) $\mathbb{F}_q$ which contains in its defining set $J = \{1,2,3,4,5,6,8,9,11,12,14,15\}$. With $k=1$, $m=2$ and $\ell = 3$, the BS bound guarantees that $d \geq m\ell + \ell = 9$. We suppose that $S_7 \neq 0$ and there is a word $c \in C$ of weight $\mathrm{w}(c) = 8$ with $\mathrm{DFT}(c) = (S_0, \ldots, S_{n-1})$. We have that $S_j = 0$ for any $j \in J$. Let us consider the generalized Newton identities for $w = 8$, which hold for any $j \geq 0$, $\sum_{i=0}^{8} S_{i+j}\sigma_{8-i} = 0$. Let us consider what the identities give for $j = k + z\ell$ with $0 \leq z \leq m-1$.

$$
\begin{aligned}
j = 1\colon &\qquad 0 = \sum_{i=0}^{8} S_{i+1}\sigma_{8-i} = \sigma_2 S_7 = 0 && \Longrightarrow \sigma_2\\
j = 4\colon &\qquad 0 = \sum_{i=0}^{8} S_{i+4}\sigma_{8-i} = \sigma_5 S_7 + \sigma_2 S_{10} = 0 && \Longrightarrow \sigma_5\\
j = 7\colon &\qquad 0 = \sum_{i=0}^{8} S_{i+7}\sigma_{8-i} = \sigma_2 S_{13} + \sigma_5 S_{10} + \sigma_8 S_7 = 0 && \Longrightarrow \sigma_8,
\end{aligned}
$$

which is a contradiction because σ_8 is the product of the locations of c, hence $\sigma_8 \neq 0$.

Theorem 7.1.8 (BS bound, reverse version)**.** *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$. Suppose that there are $m, \ell \in \mathbb{N}$, $m, \ell \geq 1$ and $k \in \{0, \ldots, n-1\}$ such that $S_k = S_{k+n}$ and for all $c \in C$:*

c) $S_{k+j+z\ell} = 0$, $j = 1 \ldots, \ell - 1$, $0 \leq z \leq m$,

d) $S_{k+(m+1)\ell+j} = 0$, $j = 1, \ldots, m\ell$.

Then:

$$d \geq m\ell + \ell\,.$$

Proof. By the BCH bound we have $d \geq m\ell + 1$. Let us suppose that there is a nonzero word $c \in C$ of weight $m\ell + 1 \leq w \leq m\ell + \ell - 1$ and locations $X_1, \ldots, X_w$. We can suppose without loss of generality that $S_{k+(m+1)\ell} \neq 0$, otherwise $S_{k+1+m\ell} = \cdots = S_{k+(m+1)\ell} = \cdots = S_{k+(2m+1)\ell} = 0$ and by the BCH bound we get $d \geq m\ell + \ell + 1 \geq m\ell + \ell$. We consider an alternative formulation of the generalized Newton identities, which is more useful for our proof:

$$\sum_{i=0}^{w} S_{w-i+j}\sigma_i = 0, \qquad \text{for any } j \geq 0, \tag{7.13}$$

where $\sigma_0 = 1$ by definition, and $\sigma_w \neq 0$. We claim that $\sigma_{z\ell} = 0$ for $0 \leq z \leq m$. If our claim is true we get a contradiction for $z = 0$, since $0 = \sigma_{0\ell} = \sigma_0 \neq 0$, and thus we prove the theorem. By induction on z, we start to prove that for $z = m$ we have $\sigma_{m\ell} = 0$. Substituting $j = k + (m+1)\ell + m\ell - w$ in (7.13):

$$\begin{aligned} 0 = \sum_{i=0}^{w} S_{w-i+j}\ \sigma_i &= \sum_{i=0}^{w} S_{k+(m+1)\ell+m\ell-i}\ \sigma_i \\ &= \sum_{i=0}^{m\ell-1} S_{k+(m+1)\ell+m\ell-i}\ \sigma_i + \sum_{i=m\ell}^{w} S_{k+(m+1)\ell+m\ell-i}\ \sigma_i \end{aligned}$$

. The right hand side of the previous equation reduces to $\sum_{i=m\ell}^{w} S_{k+(m+1)\ell+m\ell-i}\ \sigma_i$, by hypothesis d). Thus :

$$\begin{aligned} 0 &= \sum_{i=m\ell}^{w} S_{k+(m+1)\ell+m\ell-i}\ \sigma_i \\ &= S_{k+(m+1)\ell}\ \sigma_{m\ell} + \sum_{i=m\ell+1}^{w} S_{k+(m+1)\ell+m\ell-i}\ \sigma_i. \end{aligned}$$

Setting $t = (m+1)\ell - i$ we have

$$\begin{aligned} 0 &= S_{k+(m+1)\ell}\ \sigma_{m\ell} + \sum_{t=(m+1)\ell-w}^{\ell-1} S_{k+m\ell+t}\ \sigma_{(m+1)\ell-t} \\ &= S_{k+(m+1)\ell}\ \sigma_{m\ell}, \end{aligned}$$

where in the last equation we have used hypothesis a). Thus we conclude $\sigma_{m\ell} = 0$, since $S_{k+(m+1)\ell} \neq 0$, by assumption. We suppose that $\sigma_{z\ell} = 0$ for $m \geq z > \bar{z} \geq 0$ and we prove $\sigma_{\bar{z}\ell} = 0$. Substituting $j = k + (m+1)\ell + \bar{z}\ell - w$ in (7.13), we get:

$$\begin{aligned} 0 &= \sum_{i=0}^{w} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i \\ &= \sum_{i=0}^{\bar{z}\ell} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i + \sum_{i=\bar{z}\ell+1}^{w} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i \\ &= S_{k+(m+1)\ell}\ \sigma_{\bar{z}\ell} + \sum_{i=0}^{\bar{z}\ell-1} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i + \sum_{i=\bar{z}\ell+1}^{w} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i. \end{aligned} \tag{7.14}$$

We denote by A the summation $\sum_{i=0}^{\bar{z}\ell-1} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i$ and by B the summation $\sum_{i=\bar{z}\ell+1}^{w} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i$ and we prove they are zero. Setting $j = \bar{z}\ell - i$ in A, we get:

$$A = \sum_{i=0}^{\bar{z}\ell-1} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i = \sum_{j=1}^{\bar{z}\ell} S_{k+(m+1)\ell+j}\ \sigma_{\bar{z}\ell-j} = 0,$$

since $S_{k+(m+1)\ell+j} = 0$ for $1 \leq j \leq \bar{z}\ell$, by hypothesis.
Considering B we have:

$$\begin{aligned} B &= \sum_{i=\bar{z}\ell+1}^{w} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i \\ &= \sum_{i=\bar{z}\ell+1}^{m\ell} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i + \sum_{i=m\ell+1}^{w} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i \\ &= \sum_{t=\bar{z}}^{m-1} \sum_{i=t\ell+1}^{(t+1)\ell} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i + \sum_{i=m\ell+1}^{w} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i \qquad (7.15) \end{aligned}$$

For $m\ell + 1 \leq i \leq w$ we have that $k + \bar{z}\ell + 1 \leq k + (m+1)\ell + \bar{z}\ell - i \leq k + \bar{z}\ell + \ell - 1$, so $S_{k+(m+1)\ell+\bar{z}\ell-i} = 0$, by hypothesis, and (7.15) becomes:

$$\begin{aligned} B &= \sum_{t=\bar{z}}^{m-1} \sum_{i=t\ell+1}^{(t+1)\ell} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i \\ &= \sum_{t=\bar{z}}^{m-1} \left(\sum_{i=t\ell+1}^{t\ell+\ell-1} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i + S_{k+(m-t)\ell+\bar{z}\ell}\ \sigma_{t\ell+\ell} \right). \end{aligned}$$

By inductive hypothesis $\sigma_{t\ell+\ell} = 0$ for $\bar{z} \leq t \leq m-1$, hence:

$$\begin{aligned} B &= \sum_{t=\bar{z}}^{m-1} \sum_{i=t\ell+1}^{t\ell+\ell-1} S_{k+(m+1)\ell+\bar{z}\ell-i}\ \sigma_i. \\ &= \sum_{t=\bar{z}}^{m-1} \sum_{j=1}^{\ell-1} S_{k+(m+1)\ell+(\bar{z}-t)\ell-j}\ \sigma_i \quad \text{setting } j = i - t\ell \\ &= \sum_{t=\bar{z}}^{m-1} \sum_{i=1}^{\ell-1} S_{k+(m+\bar{z}-t)\ell+i}\ \sigma_i = 0 \quad \text{setting } i = \ell - j, \end{aligned}$$

because $\bar{z}+1 \leq m+\bar{z}-t \leq m$ for $\bar{z} \leq t \leq m-1$ and $1 \leq i \leq \ell-1$, so, by hypothesis, $S_{k+(m+\bar{z}-t)\ell+i} = 0$. Thus (7.14) becomes $S_{k+(m+1)\ell}\ \sigma_{\bar{z}\ell} = 0$, which implies $\sigma_{\bar{z}\ell} = 0$, since $S_{k+(m+1)\ell} \neq 0$, by hypothesis. □

Example 7.1.9. Let C be an $[n, k, d]$ cyclic codes of length $n \geq 32$ over (any) $\mathbb{F}_q$ which contains in its defining set

$$J = \{6, 7, 8, 10, 11, 12, , 14, 15, 16, 18, 19, 20, 22, 23, \ldots, 33\}.$$

With $k = 5$, $m = 3$ and $\ell = 4$, the BS bound guarantees that $d \geq m\ell + \ell = 16$. We suppose that $S_{21} \neq 0$ and that there is a word $c \in C$ of weight $\mathrm{w}(c) = 15$ with $\mathrm{DFT}(c) = (S_0, \ldots, S_{n-1})$. We have that $S_j = 0$ for any $j \in J$. Let us consider the generalized Newton identities for $w = 15$, which hold for any $j \geq 0$, $\sum_{i=0}^{15} S_{15-i+j}\sigma_i = 0$. Let us consider what the identities give for $j = k+m\ell+1+z\ell$ with $0 \leq z \leq m-1$.

$$\begin{aligned}
j = 18\colon \quad & 0 = \sum_{i=0}^{15} S_{15-i+18}\sigma_{15-i} = S_{21}\sigma_{12} = 0 && \Longrightarrow \sigma_{12}\\
j = 14\colon \quad & 0 = \sum_{i=0}^{15} S_{15-i+14}\sigma_{15-i} = S_{21}\sigma_8 + S_{17}\sigma_{12} = 0 && \Longrightarrow \sigma_8\\
j = 10\colon \quad & 0 = \sum_{i=0}^{15} S_{15-i+10}\sigma_{15-i} = S_{21}\sigma_4 + S_{17}\sigma_8 + S_{13}\sigma_{12} = 0 && \Longrightarrow \sigma_4\\
j = 6\colon \quad & 0 = \sum_{i=0}^{15} S_{15-i+6}\sigma_{15-i} = S_{21}\sigma_0 + S_{17}\sigma_4 + S_{13}\sigma_8 + S_9\sigma_{12} = 0 && \Longrightarrow \sigma_0,
\end{aligned}$$

which is a contradiction because $\sigma_0 = 1$, by definition.

7.1.4 A polynomial interpretation of Boston's bounds

Here, we consider a slight generalization of the bounds presented by Boston, as done in Remark 3.4.24. As usual, we first provide the spectral version of Boston's bound and then we give a proof, using the generalized Newton identities.

Theorem 7.1.10 (Boston bound I, gen.)**.** *Let C be an $[n, k, d]$ code over $\mathbb{F}_q$. Suppose that $3 \nmid n$ and that there is $k \in \{0, \ldots, n-1\}$ such that for all $c \in C$:*

$$S_k = S_{k+1} = S_{k+3} = S_{k+4} = 0$$

Then:

$$d \geq 4.$$

Proof. It is a special case of Theorem 7.1.3, with $\ell = k$, $m = 2$, $r = 1$ and $s = 2$. □

Theorem 7.1.11 (Boston bound II, gen.)**.** *Let C be an $[n, k, d]$ code over $\mathbb{F}_q$. Suppose that $3 \nmid n$ and that there is $k \in \{0, \ldots, n-1\}$ such that for all $c \in C$:*

$$S_k = S_{k+1} = S_{k+3} = S_{k+5} = 0$$

Then:

$$d \geq 4.$$

Proof. It is a special case of Theorem 7.1.6, with $\ell = 2$ and $m = 1$. □

Theorem 7.1.12 (Boston bound III, gen.). *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$. Suppose that $3 \nmid n$ and that there is $k \in \{0,\dots,n-1\}$ such that for all $c \in C$:*

$$S_k = S_{k+1} = S_{k+3} = S_{k+4} = S_{k+6} = 0$$

Then:

$$d \geq 5.$$

Proof. By the BCH bound we have that $d \geq 3$, so we only have to see that there are no words of weight 3 or 4 to prove the theorem. We can suppose that S_{k+2} and S_{k+5} are different from zero, otherwise by the BCH bound the claim is satisfied. In the same way, we can also suppose that $S_{k+7} \neq 0$, otherwise by the HT bound with $m = 2$ and $r = 1$, we have $d \geq 5$. Let us suppose that there exists a word of weight 3. Writing the (7.1) for $w = 3$, $j = k+1$ and $j = k+4$, we get:

$$\begin{aligned}
j = k+1: &\qquad 0 = \sum_{i=0}^{3} S_{3-i+k+1}\sigma_i = \sigma_2 S_{k+2} &&\Longrightarrow \sigma_2 = 0\\
j = k+4: &\qquad 0 = \sum_{i=0}^{3} S_{3-i+k+4}\sigma_i = S_{k+7} + \sigma_2 S_{k+5} &&\Longrightarrow S_{k+7} = 0,
\end{aligned}$$

which is a contradiction, since we supposed $S_{k+7} \neq 0$. Similarly, if there is a word $c \in C$ of weight $w = 4$, we can write (7.1) for $w = 4$, $j = k$ and $j = k+3$, obtaining:

$$\begin{aligned}
j = k: &\qquad 0 = \sum_{i=0}^{4} S_{4-i+k}\ \sigma_i = \sigma_2 S_{k+2} &&\Longrightarrow \sigma_2 = 0\\
j = k+3: &\qquad 0 = \sum_{i=0}^{4} S_{4-i+k+3}\ \sigma_i = S_{k+7} + \sigma_2 S_{k+5} &&\Longrightarrow S_{k+7} = 0.
\end{aligned}$$

Thus the claim is proved □

Theorem 7.1.13 (Boston bound IV, gen.). *Let C be an $[n,k,d]$ code over $\mathbb{F}_q$. Let c be any word of C and* $\mathrm{DFT}(c) = (S_0,\dots,S_{n-1})$ *its* DFT *with respect to α, a fixed $n-th$ root of unity over $\mathbb{F}_q$. Suppose that $4 \nmid n$ and that there is $k \in \{0,\dots,n-1\}$ such that for all $c \in C$:*

$$S_k = S_{k+1} = S_{k+2} = S_{k+4} = S_{k+5} = S_{k+6} = S_{k+8} = 0$$

Then:

$$d \geq 6.$$

Proof. By the BCH bound we have that $d \geq 4$, so we have only to see that there are no words of weight 4 or 5. We suppose S_{k+3} and S_{k+7} are different from zero, otherwise by the BCH bound the claim is satisfied. In the same way, we can also suppose that at least one between S_{k+9} and S_{k+10} is different from zero, otherwise by the HT bound with $m = 3$ and $r = 1$, we have $d \geq 6$. Let us suppose that there exists a word of weight 4. Writing the (7.1) for $w = 4$, $j = k+1,\ k+2,\ k+5,\ k+6$, we get:

$$\begin{aligned}
j = k+1 &: \quad 0 = \sum_{i=0}^{4} S_{4-i+k+1}\ \sigma_i = \sigma_2 S_{k+3} && \Longrightarrow\ \sigma_2 = 0 \\
j = k+2 &: \quad 0 = \sum_{i=0}^{4} S_{4-i+k+2}\ \sigma_i = \sigma_3 S_{k+3} && \Longrightarrow\ \sigma_3 = 0 \\
j = k+5 &: \quad 0 = \sum_{i=0}^{4} S_{4-i+k+5}\ \sigma_i = S_{k+9} + \sigma_2 S_{k+7} && \Longrightarrow\ S_{k+9} = 0 \\
j = k+6 &: \quad 0 = \sum_{i=0}^{4} S_{4-i+k+6}\ \sigma_i = S_{k+10} + \sigma_3 S_{k+7} && \Longrightarrow\ S_{k+10} = 0
\end{aligned}$$

Similarly, if there is a word $c \in C$ of weight $w = 5$, we can write (7.1) for $w = 5$ and $j = k,\ k+1,\ k+4,\ k+5$, obtaining:

$$\begin{aligned}
j = k &: \quad 0 = \sum_{i=0}^{5} S_{5-i+k}\ \sigma_i = \sigma_2 S_{k+3} && \Longrightarrow\ \sigma_2 = 0 \\
j = k+1 &: \quad 0 = \sum_{i=0}^{5} S_{5-i+k+1}\ \sigma_i = \sigma_3 S_{k+3} && \Longrightarrow\ \sigma_3 = 0 \\
j = k+4 &: \quad 0 = \sum_{i=0}^{5} S_{5-i+k+4}\ \sigma_i = S_{k+9} + \sigma_2 S_{k+7} && \Longrightarrow\ S_{k+9} = 0 \\
j = k+5 &: \quad 0 = \sum_{i=0}^{5} S_{5-i+k+5}\ \sigma_i = S_{k+10} + \sigma_3 S_{k+7} && \Longrightarrow\ S_{k+10} = 0.
\end{aligned}$$

Thus we proved that $S_{k+9} = S_{k+10} = 0$, which is a contradiction, since we supposed that at least one between S_{k+9} or S_{k+10} is different from zero. □

7.2 Comments and further research

In the proof Theorem 8.1.6, Theorem 8.1.8 (and the easy results Theorem{0,...,10} 7.1.10, 7.1.11, 7.1.12, 7.1.13) we applied the (generalized) Newton identities directly to obtain the contradiction proving our claim. These identities come from an easy manipulation of the plain locator polynomial. Note that if the word has weight at most w, the locator has degree w, its roots contains the locations and actually its roots are exactly the locations if the weight is exactly w. The contradiction we are aiming at in these proofs is to show that such word is actually the zero word (or equivalently, that all its syndromes are zero). This argument is not new, since it has been applied in [HT72] to prove the BCH bound, although our application to the presented cases is.

The Hartmann-Tzeng bound in its restricted version ([HT72]) cannot be proved in this way, because these identities do not provide a contradiction. So in the original paper [HT72] the authors have an intuition, that is, to construct a polynomial which is a multiple of the locator. From this polynomial it is easy to derive relations similar to the Newton identities and such that they provide the desired contradiction. Although this polynomial has degree higher than w and it is bound to have parasite roots, its use is easy and the proof follows nearly mechanically. We call this polynomial the `adaptive locator` (see also [SWST96]). Unfortunately, they do not expand on this idea any further and no subsequent author has tried to develop this approach. Indeed, to prove the more general form of the HT bound, Roos in [Roo82] abandons the polynomial approach and provide proofs based on suitable matrices. What we do in Theorem 7.1.3 to prove the more general form of the HT bound is to use the adaptive locator (a multiple of the locator) of [HT72] and then derive again some special relations (similar to the Newton identities) that lead to the desired contradiction.

The above discussion allows us to conjecture the following:

- given a defining set and a length (without knowing the field), it is possible to derive an adaptive locator;
- from the adaptive locator, relations similar to the Newton identities come directly and lead to a contradiction;
- the computation of the contradiction from the adaptive locator is polynomial-time (in the length);
- the computation of the adaptive locator from (n, S) may be polynomial-time (in the length).

We find it a very interesting research problem to investigate this approach further. Should these conjectures be proved (including the fourth of which we are not completely confident), we would have that the computation of the optimal root bound f is polynomial-time.

Note that at this stage of the thesis, we have not claimed anything on the complexity of computing f and indeed its computation might even need infinite steps. However, in the next chapter we will prove that f can be computed in a finite time.

Computing the optimal root bound via Gröbner bases

In Section 3.1 we discussed the problem to compute the optimal root function, f (resp. the optimal root bound, $\mathsf{f}_{\mathcal{D}}$), in a finite time. This is a natural question, since the characterization we gave of the optimal root function both using Definition 3.1.16 and using (3.4) apparently requires an infinite number of computations. In this chapter, we show that f may be computed in a finite time, using the systems of polynomials J_C, introduced in Section 5.2.

8.1 Preliminaries and notation

We denote by $\mathcal{P}$ the subset of $\mathbb{N}$ formed by all prime numbers, $\mathcal{P} = \{2, 3, 5, \dots\}$. Given an integer $n \geq 2$, we denote by $\mathcal{P}_n$ the subset of $\mathcal{P}$ formed by all p such that $(p, n) = 1$. Let $\mathbb{K}$ be a field, not necessary finite. In the case $\mathbb{K}$ is finite, we use $\mathbb{F}_q$ to indicate the finite field with q elements. We denote by $1_{\mathbb{K}}$ the multiplicative neutral element of $\mathbb{K}$, by $0_{\mathbb{K}}$ the additive neutral element of $\mathbb{K}$, by $\mathrm{char}(\mathbb{K})$, the characteristic of $\mathbb{K}$ and by $\overline{\mathbb{K}}$ the algebraic closure of $\mathbb{K}$. We recall that the `prime field` of $\mathbb{K}$ is the smallest subfield of $\mathbb{K}$ containing $1_{\mathbb{K}}$; we denote such field with $\mathbb{P}(\mathbb{K})$. It is well-known that the prime field of $\mathbb{K}$ depends only on $\mathrm{char}(\mathbb{K})$, as in the following proposition.

Proposition 8.1.1. *Let $\mathbb{K}$ be a field. Then*

- *$\mathbb{P}(\mathbb{K})$ is $\mathbb{Q}$ if and only if $\mathrm{char}(\mathbb{K}) = 0$,*
- *$\mathbb{P}(\mathbb{K})$ is $\mathbb{F}_p$ if and only if $\mathrm{char}(\mathbb{K}) = p$ for a $p \in \mathcal{P}$.*

We denote by $\mathbb{D}(\mathbb{K})$ the `prime domain` of $\mathbb{K}$, i.e. the smallest subring of $\mathbb{K}$ containing $1_{\mathbb{K}}$. We observe that $\mathbb{D}(\mathbb{K}) = \mathbb{F}_p$ if $\mathrm{char}(\mathbb{K}) = p$ for some $p \in \mathcal{P}$, and that $\mathbb{D}(\mathbb{K}) = \mathbb{Z}$ if $\mathrm{char}(\mathbb{K}) = 0$. If $\mathbb{K}$ is understood we write $\mathbb{D} = \mathbb{D}(\mathbb{K})$ and $\mathbb{P} = \mathbb{P}(\mathbb{K})$. Let $r \geq 1$, we consider a set of variables $X = \{x_1, \dots, x_r\}$, $\mathcal{M} = \mathcal{M}(X)$ is the set of all monomials in X, $\mathbb{K}[x_1, \dots, x_r]$ is a polynomial ring over $\mathbb{K}$ with a monomial order $<$, which from now on is understood. As usual, we denote by $\mathrm{LT}(g)$ the leading term of any $g \in \mathbb{K}[x_1, \dots, x_r]$ and by X^ν the monomial $X^\nu = x_1^{\nu_1} \dots x_r^{\nu_r}$, with $\nu = (\nu_1, \dots, \nu_r) \in \mathbb{N}^r$. Note that the definition of X^ν does not depend on the field. With an abuse of notation, for any field $\mathbb{K}$ we will view X^ν as an element of $\mathbb{K}[x_1, \dots, x_r]$, when it is appropriate and convenient to us. We define a kind of Gröbner basis which we call `domain-reduced`.

Definition 8.1.2. *Let $\mathbb{K}$ be a field, I be an ideal in $\mathbb{K}[x_1, \dots, x_r]$, G be a Gröbner basis of I . We say that G is* `domain-reduced (d-red)` *if:*

1. *for any $g \in G$, any monomial X^ν of g and any $g' \in G \backslash \{g\}$, we have $\mathrm{LT}(g') \nmid X^\nu$,*
2. *any coefficient of any $g \in G$ lies in $\mathbb{D}(\mathbb{K})$,*
3. *if $\mathrm{char}(\mathbb{K}) = p$ for some $p \in \mathcal{P}$, then any g is monic,*
4. *if $\mathrm{char}(\mathbb{K}) = 0$, then $\mathrm{LC}(g) > 0$ and for any $g \in G$ there is no integer $n \geq 2$ such that n divides all the coefficients of g.*

We note the two following obvious facts.

Fact 8.1.3. *Let I be an ideal in $\mathbb{K}[X]$. Let G' be the reduced Gröbner basis of I. Suppose that G is a d-red Gröbner basis for I. Then $\{\mathrm{LM}(G')\} = \{\mathrm{LM}(G)\}$. Moreover, for any $g \in G$ there is a $g' \in G'$ such that $g' = \lambda g$, with $\lambda \in \mathbb{K}$.*

Fact 8.1.4. *If $\mathrm{char}(\mathbb{K}) = p$ for some $p \in \mathcal{P}$, then*

$$G \textit{ is d-red} \iff G \textit{ is a reduced Gröbner basis and } \forall\ g \in G,\ g \in \mathbb{D}[X] = \mathbb{P}[X].$$

We observe that not all ideals in $\mathbb{K}[x_1, \dots, x_r]$ have a d-red Gröbner basis, as the next example shows.

Example 8.1.5. Let $\mathbb{K}[x_1, \dots, x_r] = \mathbb{F}_4[x]$, with $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$ and $\alpha^2 = \alpha + 1$. We consider the ideal $I = \langle x - \alpha \rangle$ and we claim that it does not have a d-red Gröbner basis. In fact, if a d-red basis exists for I, it means that $I = \langle g(x) \rangle$, with $g(x) \in \mathbb{F}_2[x]$ and such that $\mathrm{LT}(g) \mid \mathrm{LT}(x - \alpha) = x$, thus $\mathrm{LT}(g) = x$. The only two polynomials in $\mathbb{F}_2[x]$ with leading term x are $g_1(x) = x$ and $g_2(x) = x + 1$, but none of them belongs to I, as it is easy to check. Thus a d-red basis for I does not exist.

Example 8.1.6. Let $\mathbb{K}[x_1, \dots, x_r] = \mathbb{R}[x]$. We consider the ideal $I = \langle x - \sqrt{2} \rangle$ and we claim that it does not have a d-red Gröbner basis. In fact, if a d-red basis exists for I, it means that $I = \langle g(x) \rangle$, with $g(x) \in \mathbb{Z}[x]$ and such that $\mathrm{LT}(g) \mid \mathrm{LT}(x - \sqrt{2}) = x$, thus $\mathrm{LT}(g) = x$. But such g cannot exist, since the minimal polynomial of $\sqrt{2}$ in $\mathbb{Z}[x]$ is $x^2 - 2$. Thus a d-red basis for I does not exist.

Nevertheless, if for an ideal $I \subseteq \mathbb{K}[x_1,\dots,x_r]$ a d-red Gröbner basis exists, then it is unique.

Lemma 8.1.7. *Let I be an ideal in $\mathbb{K}[x_1,\dots,x_r]$ such that $G = \{g_1,\dots,g_t\}$ is a d-red Gröbner basis of I. Then G is unique.*

Proof. If $\mathrm{char}(\mathbb{K}) = p$ for some $p \in \mathcal{P}$, by Fact 8.1.4 we have that $G = \mathrm{GB}(I)$ is the reduced bases of I, thus it is unique.

If $\mathrm{char}(\mathbb{K}) = 0$, then $g_i \in \mathbb{Z}[x_1,\dots,x_r]$ for $1 \le i \le t$. Let G' be another d-red basis for I. From Fact 8.1.3 we have that G' shares the leading monomials with G, so $G' = \{\lambda_1 g_1,\dots,\lambda_t g_t\}$ for some non-zero $\lambda_1,\dots,\lambda_t$ in $\mathbb{Z}$. Let $1 \le i \le t$. Since G and G' are d-red bases, we have $\mathrm{LC}(g_i) > 0$ and $\mathrm{LC}(\lambda_i g_i) = \lambda_i \mathrm{LC}(g_i) > 0$, which imply $\lambda_i > 0$. On the other hand, λ_i is a positive integer which divides all coefficients of $\lambda_i g_i$, so, for 4. of Definition 8.1.2, $\lambda_i = 1$. But then $G = G'$. □

Since the d-red Gröbner basis for an ideal I is unique, if it exists, we may denote by $G(I)$ the d-red Gröbner basis of I, with the convention that $G(I) = \emptyset$ if it does not exist for I.

There are some cases where we can prove that a d-red basis exists, as for example when $\mathbb{K}$ is a prime field. Moreover, such basis can be computed from the reduced Gröbner basis, as we are going to show.

Given $a_1,\dots,a_r$ integers we denote by $\gcd(a_1,\dots,a_r)$ the greatest common divisor of $a_1,\dots,a_r$ (but we can also use the notation $\gcd(\{a_i\}_{1\le i\le r})$). Note that 4. of Definition 8.1.2 can be reformulated as: if $\mathrm{char}(\mathbb{K}) = 0$, then, for any $g \in G$, $\mathrm{LT}(g) > 0$ and $\gcd(\{a_\nu\}_{\nu\in N_g}) = 1$, where the a_ν's are the coefficients of g.

Proposition 8.1.8. *Let $\mathbb{K}$ be a prime field and let I be any ideal in $\mathbb{K}[x_1,\dots,x_r]$. Then I has a d-red Gröbner basis.*

Proof. Let $G = \mathrm{GB}(I) = \{g_1,\dots,g_t\}$ be the reduced Gröbner basis of I.

If $\mathrm{char}(\mathbb{K}) = p$ for some $p \in \mathcal{P}$, then $\mathbb{D}(\mathbb{K}) = \mathbb{K}$, so $G \subseteq \mathbb{D}[x_1,\dots,x_r]$ and by Fact 8.1.4 we have that G is also the d-red Gröbner basis of I.

Let us consider the case $\mathrm{char}(\mathbb{K}) = 0$, i.e. $\mathbb{K} = \mathbb{Q}$. By definition of reduced Gröbner basis (Definition 5.1.5), for $1 \le i \le t$ we have $g_i = X^{\mu_i} + \sum_{\nu\in N_i} \alpha_\nu X^\nu$ for some finite subset $N_i \subseteq \mathbb{N}^r$, where $\mathrm{LT}(g_i) = X^{\mu_i}$ and $\alpha_\nu = a_\nu/b_\nu$ with a_ν, $b_\nu \in \mathbb{Z}$, $(a_\nu, b_\nu) = 1$ and $b_\nu \ge 1$. We also write $a_{\mu_i} = b_{\mu_i} = 1$. For $1 \le i \le t$ let us take $\ell_i \in \mathbb{Q}$ defined by

$$\ell_i = \frac{\prod_{\nu\in N_i} b_\nu}{\gcd(\{b_\nu\}_{\nu\in N_i})}.$$

Since ℓ_i is the least common multiple of the b_ν's, if $\tilde{\ell}_i$ is any integer such that $b_\nu \mid \tilde{\ell}_i$ for any $\nu \in N_i$, then $\ell_i \mid \tilde{\ell}_i$. By construction, we have that $\ell_i g_i$ is in $\mathbb{Z}[x_1,\dots,x_r]$ and actually $\ell_i g_i$ and g_i have the same monomials. Moreover, for all i it is easy to see that $\gcd(\{\ell_i a_\nu / b_\nu\}_{\nu \in N_i \cup \{\mu_i\}}) = 1$. So the basis $G' = \{\ell_1 g_1, \dots, \ell_t g_t\}$ is a d-red Gröbner basis of I. □

Note that in the proof of Proposition 8.1.8 we do not need that $\mathbb{K}$ is a prime field, since the only thing we need is that the reduced Gröbner basis of I belongs to $\mathbb{P}(\mathbb{K})[X]$. This allows us to state the following result.

Proposition 8.1.9. *Let I be any ideal in $\mathbb{K}[x_1, \dots, x_r]$ such that its reduced Gröbner basis, $G = \mathrm{GB}(I) \subseteq \mathbb{P}[X]$. Then I has a d-red Gröbner basis.*

In the previous proof we state that, given $a_1, \dots a_n$ and $b_1, \dots, b_n$ such that $(a_i, b_i) = 1$ for $1 \leq i \leq n$, if we define $d = \gcd(\{b_1, \dots, b_n\})$ and $\ell = \prod_{i=1}^n b_i/d$, then $\gcd(\ell, \ell a_1/b_1, \dots, \ell a_n/b_n) = 1$. We provide here a proof of this fact for $n = 2$, the other cases follow in a similar way, by induction.

Example 8.1.10. Let us consider a_1, a_2, b_1, $b_2 \in \mathbb{Z}$ with $(a_1, b_1) = 1$ and $(a_2, b_2) = 1$. We denote by $d = (b_1, b_2)$ the greatest common divisor of b_1 and b_2 and with $\ell = [b_1, b_2]$ their least common multiple. We have:

$$b_1 = dt_1, \qquad b_2 = dt_2, \qquad \ell = dt_1 t_2,$$

for some $t_1, t_2 \in \mathbb{Z}$ with $(t_1, t_2) = 1$. We want to prove that $\gcd(\ell, \ell a_1/b_1, \ell a_2/b_2) = 1$. Noting that $\ell a_1/b_1 = a_1 t_2$ and $\ell a_2/b_2 = a_2 t_1$, we have:

$$\begin{aligned} \gcd(\ell, \ell a_1/b_1, \ell a_2/b_2) &= \gcd(dt_1t_2, a_1t_2, a_2t_1) \\ &= ((dt_1t_2, a_1t_2), a_2t_1) \\ &= ((b_1t_2, a_1t_2), a_2t_1) \\ &= (t_2(a_1, b_1), a_2t_1) \\ &= (t_2, a_2t_1) \\ &= (t_2, a_2) = 1 \end{aligned}$$

where the last equality is due to $1 = (a_2, b_2) = (a_2, dt_2)$.

An example of ideals for which a d-red Gröbner basis obviously exists are those ideals with empty variety, since in this case $\mathrm{GB}(I) = G(I) = \{1\}$:

Corollary 8.1.11. *Let $\mathbb{K}$ be any field and I be an ideal in $\mathbb{K}[x_1, \dots, x_r]$. Then:*

$$\mathcal{V}(I) = \emptyset \iff \mathrm{GB}(I) = \{1\} \iff G(I) = \{1\}.$$

8.2 Linked ideals

Let N be a finite subset of $\mathbb{N}^r$ and let $F = \{f_1, \dots, f_t\}$ be any finite set of functions

$$f_i \colon N \to \mathbb{Z}.$$

We will call F a set of `defining functions` (see `function ring` in [Rei06]). We view N as set of indices for a finite subset of monomials in $\mathcal{M}(X)$, where the field is unspecified. From F, for any field $\mathbb{K}$, we want to construct an ideal in $\mathbb{K}[X]$.

Definition 8.2.1. *Let $\mathbb{K}$ be a field. Let N be a finite subset of $\mathbb{N}^r$ and $F = \{f_1, \dots, f_t\}$ a set of defining functions. For any $\nu \in N$ we consider a monomial X^ν in $\mathcal{M}$ as an element of $\mathbb{K}[X]$. We define a map $\psi \colon \mathbb{Z} \mapsto \mathbb{K}$ by:*

$$\psi(n) = \begin{cases} 0_{\mathbb{K}} & \text{if } n = 0 \\ \underbrace{1_{\mathbb{K}} + \cdots + 1_{\mathbb{K}}}_{n-times} & \text{if } n > 0 \\ -(\underbrace{1_{\mathbb{K}} + \cdots + 1_{\mathbb{K}}}_{(-n)-times}) & \text{if } n < 0. \end{cases}$$

We denote by $\mathcal{I}(\mathbb{K}, F)$ the ideal in $\mathbb{K}[X]$ generated by $\mathsf{q}^{\mathbb{K}}(F) = \{\mathsf{q}^{\mathbb{K}}(f_1), \dots, \mathsf{q}^{\mathbb{K}}(f_t)\}$, where for any $1 \leq i \leq t$, $\mathsf{q}^{\mathbb{K}}(f_i)$ is a polynomial in $\mathbb{D}[X]$

$$\mathsf{q}^{\mathbb{K}}(f_i) = \sum_{\nu \in N} \psi(f_i(\nu)) X^\nu, \ .$$

Observe that, for any field $\mathbb{K}$, the image of ψ is contained in $\mathbb{D}$.

Definition 8.2.2. *Let $\mathbb{K}$ be a field. Let I be an ideal in $\mathbb{K}[X]$. We say that I is* ***`simply-generated by`*** *B if there is a finite basis $B = \{b_1, \dots, b_s\}$ for I s.t. $B \subseteq \mathbb{D}[X]$. We say that I is simply-generated if it is simply-generated by B for some B.*

Lemma 8.2.3. *Let I be an ideal in $\mathbb{K}[X]$ generated by t polynomials, $g_1, \dots, g_t \in \mathbb{D}[X]$. Then it is possible to compute a Gröbner basis for I performing only operations in $\mathbb{D}[X]$.*

Proof. To obtain a Gröbner basis from $\{g_1, \dots, g_t\}$ the Möller algorithm ([BM09], [Morar], [Möl88]) prescribes two operations, which are applied iteratively on an intermediate basis B, being the first B equal to $\{g_1, \dots, g_t\}$. The first is the computation of the S-polynomials of all pairs from B. The second is the reduction of the S-polynomials with respect to B. There are several definitions for S-polynomials present in the literature. The one we use here is in accordance with [BM09] and Corollary 46.6.1 in [Morar]

$$S(a,b) = \frac{\mathrm{lcm}(\mathrm{LC}(a), \mathrm{LC}(b))}{\mathrm{LC}(a)} \frac{\mathrm{lcm}(\mathrm{LM}(a), \mathrm{LM}(b))}{\mathrm{LM}(a)} a - \frac{\mathrm{lcm}(\mathrm{LC}(a), \mathrm{LC}(b))}{\mathrm{LC}(b)} \frac{\mathrm{lcm}(\mathrm{LM}(a), \mathrm{LM}(b))}{\mathrm{LM}(b)} b$$

It is immediate that all coefficients involved in this computation remain in $\mathbb{D}$. As regards reductions, we consider the Zacharias canonical normal form reduction ([Morar], [Zac78]) which again keeps the coefficients in $\mathbb{D}$. As a consequence, all the operations performed in this algorithm will keep the coefficients in $\mathbb{D}$ and so all intermediate bases will be in $\mathbb{D}[X]$ as well. When the algorithm terminates, the last intermediate basis will be a Gröbner basis for I. □

Thanks to the previous lemma, it is easy to see that a simply-generated ideal has a d-red Gröbner basis.

Corollary 8.2.4. *Let I be a simply-generated ideal. Then:*

1. *I has a Gröbner basis $G' \subset \mathbb{D}[X]$.*
2. *its reduced Gröbner basis $G = \mathrm{GB}(I)$ is in $\mathbb{P}[X]$*
3. *it has a d-red Gröbner basis $G'' = G(I)$*

Proof.

1. Thanks to Lemma 8.2.3, we have computed a Gröbner basis G' for I performing only operations in $\mathbb{D}[X]$ and so $G' \subset \mathbb{D}[X]$.
2. From G' we can easily obtain the reduced Gröbner basis G by performing interreductions in G. Since the interreductions involve only polynomial divisions and we start from polynomials over the field $\mathbb{P}$, the resulting polynomial set will be again in $\mathbb{P}[X]$.
3. It is a direct consequence of 2) and Proposition 8.1.9.

□

Note that, generally speaking, $G' \neq G \neq G''$.

Definition 8.2.5. *Let $\mathcal{K} = \{\mathbb{K}_h\}_{h\in H}$ be a set of fields, indexed by a set H. For any $h \in H$, let I_h be an ideal in $\mathbb{K}_h[x_1, \dots, x_r]$.*

We say that the ideal set $\{I_h\}_{h\in H}$ is F-`linked` if there is a set of defining function $F = \{f_1, \dots, f_t\}$ with $f_i \colon N \mapsto \mathbb{Z}$, N finite subset of $\mathbb{N}^r$, $1 \leq i \leq t$, such that:

$$\text{for any } h \in H, \quad I_h = \mathcal{I}(\mathbb{K}_h, F).$$

Example 8.2.6. We take $X = \{x, y\}$, $\mathbb{K}_1 = \mathbb{F}_2$, $\mathbb{K}_2 = \mathbb{F}_3$. Let $I_2 = \langle x, x+y+xy, x+y\rangle$ be an ideal in $\mathbb{F}_2[x, y]$ and $I_3 = \langle 2x+y+2xy, 2x+y\rangle$ be an ideal in $\mathbb{F}_3[x, y]$. To see that I_2 and I_3 are $F-$linked we consider the ideal I_0 in $\mathbb{Q}[x, y]$, defined by $I_0 = \langle g_1, g_2, g_3\rangle$

with $g_1 = 5x+4y+2xy$, $g_2 = 3x+3y+3xy$, $g_3 = -x+y$. By reducing the coefficients of g_1, g_2, g_3 in $\mathbb{F}_2$ we obtain $g_1' = x$, $g_2' = x+y+xy$, $g_3' = x+y$, which is a basis for I_2. Similarly, by reducing the coefficients of g_1, g_2, g_3 in $\mathbb{F}_3$ we obtain $g_1'' = 2x+y+2xy$, $g_2'' = 0$, $g_3'' = 2x+y$, which is a basis for I_3. Thus I_2 and I_3 are F-linked, choosing $N = \{(1,0),(0,1),(1,1)\}$, $F = \{f_1, f_2, f_3\}$ with

$$\begin{array}{lll} f_1(1,0) = 5, & f_2(1,0) = 3, & f_3(1,0) = -1, \\ f_1(0,1) = 4, & f_2(0,1) = 3, & f_3(0,1) = 1, \\ f_1(1,1) = 2, & f_2(1,1) = 3, & f_3(1,1) = 0. \end{array}$$

Lemma 8.2.7. *Let $\mathcal{K} = \{\mathbb{K}_h\}_{h\in H}$ be a set of fields, indexed by a set H. For any $h \in H$, let I_h be an ideal in $\mathbb{K}_h[X]$. If $\{I_h\}_{h\in H}$ is an F-linked set, then any $I \in \{I_h\}_{h\in H}$ is simply-generated.*

Proof. Let us suppose $F = \{f_1, \dots, f_t\}$. Let $h \in H$. Let $\mathbb{F} = \mathbb{K}_h$. We have that $I = I_h$. By definition $I = \mathcal{I}(\mathbb{F}, F) = \langle \mathsf{q}^{\mathbb{F}}(f_1), \dots \mathsf{q}^{\mathbb{F}}(f_t)\rangle$, where $\mathsf{q}^{\mathbb{F}}(f_i) = \sum_{\nu\in N} \psi(f_i(\nu))X^\nu$, $1 \le i \le t$ and $\psi(f_i(\nu)) \in \mathbb{D}(\mathbb{F})$ for any $\nu \in N$. Thus I is simply-generated. □

Lemma 8.2.8. *Let $I \subseteq \mathbb{K}[x_1, \dots, x_r]$ be a simply-generated ideal. Then there is a set of defining functions F such that the set $\{I\}$ is $F-$linked.*

Proof. Let $B = \{b_1, \dots, b_s\}$ be a finite basis for $I \subseteq \mathbb{K}[x_1, \dots, x_r]$ s.t.

$$b_j = \sum_{\nu\in N_j} a_{\nu,j} X^\nu, \quad 1 \le j \le s$$

with any coefficient $a_{\nu,j}$ in $\mathbb{D}$. This basis must exist because I is simply-generated. Let N be $\cup_{1\le j\le s} N_j$. For any $1 \le j \le s$, we construct a function $f_j \colon N \mapsto \mathbb{Z}$, as follows:

$$f_j(\nu) = \begin{cases} a_{\nu,j}, & \text{if } \nu \in N_j \\ 0, & \text{otherwise.} \end{cases}$$

Let F be the set $\{f_1, \dots, f_s\}$. Then by construction it is obvious that

$$I = \mathcal{I}(\mathbb{K}, F).$$

□

In the previous lemma, we have seen how to define a function set from a basis B for simply-generated ideal. From now on, we denote by $\mathcal{F}(B)$ the function set so obtained. So, the lemma could be made more precise by stating that I is $\mathcal{F}(B)$-linked. The definition of $\mathcal{F}(B)$-linked set depends clearly on the choice of the basis B.

Example 8.2.9. Let us consider the basis $B = \{2y + 1, x\}$ in $\mathbb{Z}[x, y]$ and the ideals $I_0 = \mathcal{I}(\mathbb{Q}, \mathcal{F}(B))$ and $I_2 = \mathcal{I}(\mathbb{F}_2, \mathcal{F}(B))$. Clearly, $G_0 = \{2y + 1, x\}$ is the d-red Gröbner basis of I_0 with respect any monomial order, while the d-red basis of I_2 is $G_2 = \{1\}$. So, I_0 and I_2 are $\mathcal{F}(G_0)$-linked but not $\mathcal{F}(G_2)$-linked.

Example 8.2.10. Let $I_0 = \langle y+2x, y-x+z\rangle \in \mathbb{Q}[x, y, z]$ and $I_3 = \langle y-x, y-x+z\rangle \in \mathbb{F}_3[x, y, z]$ be two F-linked ideals. Their d-red bases with respect to the lexicographic order $z < y < x$ are, respectively, $G(I_0) = \{3y + 2z, 3x - z\}$ and $G(I_3) = \{z, x - y\}$. Clearly, I_0 and I_3 are $\mathcal{F}(B)$-linked, with $B = \{y + 2x, y - x + z\}$. We show that they are neither $\mathcal{F}(G_0)$-linked, nor $\mathcal{F}(G_3)$-linked. In fact:

- if I_0 and I_3 are $\mathcal{F}(G_0)$-linked, we have that $\{z, 2z\}$ is a basis of I_3, which is false;

- if I_0 and I_3 are $\mathcal{F}(G_3)$-linked, we have that $\{z, x - y\}$ is a basis for I_0. Note that $\{z, x - y\}$ would then be the d-red basis of I_0, which is impossible, since I_0 has the d-red basis $\{3y + 2z, 3x - z\}$.

In Definition 8.2.5, we called F-linked the ideals over different fields which have formally a same basis in $\mathbb{Z}[x_1, \ldots, x_r]$, let us say, B. In the following theorem we give a sufficient condition for which two F-linked ideals share also a same Gröbner basis in $\mathbb{Z}[x_1, \ldots, x_r]$ and we expose some consequence of this fact. To obtain this, starting from a basis in $\mathbb{Z}[x_1, \ldots, x_r]$, we need an algorithm which provides a Gröbner basis in $\mathbb{Z}[x_1, \ldots, x_r]$ performing computations only in $\mathbb{Z}[x_1, \ldots, x_r]$. This is possible thanks to the development of a Gröbner theory for polynomial rings over euclidean domains started by Kandri-Rody and Kapur in [KRK88], improved by L. Pan for polynomial rings over principal ideal domains ([Pan89]) and finally concluded by Möller for polynomial over principal ideal rings ([Möl88]). Once we can use the Möller algorithm to obtain a Gröbner basis from B, we adapt the idea of `Gröbner trace` in [Tra88] to find a set of F-linked ideals which have a Gröbner basis formally equivalent to one produced by Möller algorithm.

Theorem 8.2.11. *Let $\mathcal{K} = \{\mathbb{K}_h\}_{h\in H}$ be a set of fields, indexed by a set H, such that for any $p \in \mathcal{P}$ there exist $h_p \in H$ with $\mathrm{char}(\mathbb{K}_{h_p}) = p$. If $0 \in H$, then $\mathbb{K}_0 = \mathbb{Q}$. For any h in H let I_h be an ideal in $\mathbb{K}_h[x_1, \ldots, x_r]$. Let $<$ be any ordering. Suppose that $\{I_h\}_{h\in H}$ is F-linked. Denote by $G_h = G(I_h)$ the d-red basis of I_h w.r.t. $<$. If $0 \notin H$ let $I_0 = \mathcal{I}(\mathbb{Q}, F)$. Denote by $G_0 = G(I_0)$ the d-red basis of I_0.*

Then there is a prime $\bar{p} \in \mathcal{P}$ and $\bar{B} \subseteq \mathbb{Z}[x_1, \ldots, x_r]$ Gröbner basis of I_0, such that the ideals $\{I_h \mid h \in H, \mathrm{char}(\mathbb{K}_h) \geq \bar{p}\}$ are $\mathcal{F}(\bar{B})$-linked. Let $\bar{F} = \mathcal{F}(\bar{B})$. Then

a) $\mathsf{q}^{\mathbb{K}_h}(\bar{F})$ is a Gröbner basis for I_h if $\mathrm{char}(\mathbb{K}_h) \geq \bar{p}$;

b) for any $p \geq \bar{p}$, $G_{h_p} = \{1\} \iff G_0 = \{1\}$;

c) $\bar{p}$ can be computed in a finite time.

Proof. Since $\{I_h\}_{h\in H}$ is an F-linked set, there is a basis $B = \{b_1, \dots, b_s\}$ with $b_i \in \mathbb{Z}[x_1, \dots, x_r]$ for $1 \leq i \leq s$ such that, for any $h \in H$, $\mathsf{q}^{\mathbb{K}_h}(\mathcal{F}(B)) = \mathsf{q}^{\mathbb{K}_h}(F)$ is a basis for I_h. Thanks to Möller algorithm ([BM09, Morar, Möl88]), we can compute in $\mathbb{Z}[x_1, \dots, x_r]$ a Gröbner basis of I_0, involving only coefficients in $\mathbb{Z}$ as described in Lemma 8.2.3. Let us denote such basis as $\bar{B}$. Let C be the subset of $\mathbb{Z}$ containing all coefficients that occur in the computation of $\bar{B}$. By termination of the Möller algorithm, C is finite and hence there are two integers m_1, $m_2 \in \mathbb{Z}$ such that $m_1 = \min\{c \mid c \in C\}$ and $m_2 = \max\{c \mid c \in C\}$. Let $\bar{p}$ be the smallest prime number s.t. $\bar{p} > |m_1|, |m_2|$. Suppose now that $p \in \mathcal{P}$ is such that $p > \bar{p}$, $I_{h_p} = \mathcal{I}(\mathbb{K}_{h_p}, F)$ with $\mathrm{char}(\mathbb{K}_{h_p}) = p$, and we compute a Gröbner basis for I_{h_p} using the Möller algorithm as in the I_0 case ([Tra88]). Since I_{h_p} is F-generated, it has a basis $B_p = \mathsf{q}^{\mathbb{K}_{h_p}}(F) = \mathsf{q}^{\mathbb{K}_{h_p}}(\mathcal{F}(B))$ which is formally the same as $B_0 = \mathsf{q}^{\mathbb{Q}}(F) = B$, but now when we make calculations we have to reduce modulo p every time we compute a new coefficient. But we <u>never need to do so</u>, because p is larger than any coefficient which appears in our computation. Hence every calculation in the I_{h_p} case is formally the same as the corresponding calculation in I_0 case. In particular, the resulting basis will be the same, from a formal point of view, so that the two ideals are $\mathcal{F}(\bar{B})$-linked.

a) It is clear that $\mathsf{q}^{\mathbb{K}_{h_p}}(\bar{F})$ is a Gröbner basis of I_{h_p}, since the reduction of the S-polynomials of $\mathsf{q}^{\mathbb{K}_{h_p}}(\bar{F})$ is zero, as it is possible to check following the corresponding computations for the S-polynomials of $\bar{B}$.

b) We have that $G_0 = \{1\}$ if and only if $\bar{B}$ contains a constant polynomial different from zero. In the same way, thanks to a), $G_{h_p} = \{1\}$ if and only if $\mathsf{q}^{\mathbb{K}_{h_p}}(\bar{F})$ contains a constant polynomial different from zero. By construction, $\bar{B}$ contains a constant polynomial different from zero if and only if $\mathsf{q}^{\mathbb{K}_{h_p}}(\bar{F})$ contains a constant polynomial different from zero. So, $G_0 = \{1\} \iff G_{h_p} = \{1\}$ for any $p \geq \bar{p}$.

c) $\bar{p}$ is clearly computed in a finite time, since the coefficient set C is finite and the Möller algorithm terminates in a finite time.

□

We can use the previous lemmas, in order to prove the main result of this section.

Theorem 8.2.12. *Let $\mathcal{K} = \{\mathbb{K}_h\}_{h\in H}$ be a set of fields, indexed by a set H. For any h in H let I_h be an ideal in $\mathbb{K}_h[X]$. Suppose that the ideal set $\{I_h\}_{h\in H}$ is F-linked.*

Suppose that we want to test whether there is an h s.t. $\mathcal{V}(I_h) \neq \emptyset$. Then we can perform our test in a finite time.

Proof. We assume without loss of generality $F = \{f_1, \dots, f_t\}$. Let $<$ be any ordering and for any $h \in H$, let $G_h = G(I_h)$ be the d-red basis of I_h w.r.t. $<$.

We have already seen that $\mathcal{V}(I_h) = \emptyset$ if and only if $G_h = \{1\}$. Therefore, in order to perform our test we would have to compute all G_h for $h \in H$. If H is finite, we have our claim. We suppose that H is infinite. We start to enlarge $\{\mathbb{K}_h\}_{h\in H}$ (and $\{I_h\}_{h\in H}$), by adding all prime fields and $\mathbb{Q}$, as follows. We denote by H' the index set

$$H' = H \cup \{0\} \cup \mathcal{P}.$$

Permuting the indices of H', if necessary, we can assume without loss of generality that $\mathbb{K}_0$ is the field of rationals $\mathbb{K}_0 = \mathbb{Q}$ and for any $p \in \mathcal{P}$, $\mathbb{K}_p$ is the prime field $\mathbb{K}_p = \mathbb{Z}_p$. We have that

$$\{\mathbb{K}_h\}_{h\in H'} = \{\mathbb{K}_h\}_{h\in H} \cup \mathbb{Q} \cup \{\mathbb{Z}_p\}_{p\in\mathcal{P}}.$$

Let I_0 be the ideal $\mathcal{I}(\mathbb{Q}, F)$ which is a simply-generated ideal in $\mathbb{Q}[x_1, \dots, x_r]$. Similarly, for any $p \in \mathcal{P}$, I_p is the ideal $\mathcal{I}(\mathbb{Z}_p, F)$ which is a simply-generated ideal in $\mathbb{Z}_p[x_1, \dots, x_r]$. By construction, we have that

$$\{I_h\}_{h\in H'} \quad \text{is an } F\text{-linked set.}$$

We decompose our field set and ideal set according to the characteristic. For any $p \in \mathcal{P}$, let H^p, H^0 s.t.

$$\begin{gathered} H' = (\sqcup_{p\in\mathcal{P}} H^p) \sqcup H^0, \\ \forall\, p \in \mathcal{P},\ \forall\, h \in H^p,\ \mathrm{char}(\mathbb{K}_h) = p, \\ \forall\, h \in H^0,\ \mathrm{char}(\mathbb{K}_h) = 0. \end{gathered}$$

Since $\{I_h\}_{h\in H'}$ is F-linked, then any I_h for $h \in H'$ is simply-generated and we denote by B_h the basis $\mathsf{q}^{\mathbb{K}_h}(F)$ of I_h, where $\mathsf{q}(f_j) \in \mathbb{D}(\mathbb{K}_h)$, for any $h \in H'$ and for all $1 \le j \le t$. Then we immediately get that for any $p \in \mathcal{P}$,

$$\forall h_1,\ h_2\ \in H^p, \qquad B_{h_1} = B_{h_2}, \qquad G_{h_1} = G_{h_2}, \tag{8.1}$$

$$\forall h_1,\ h_2\ \in H^0, \qquad B_{h_1} = B_{h_2}, \qquad G_{h_1} = G_{h_2}. \tag{8.2}$$

By Theorem 8.2.11 we have that there exists $\bar{p} \in \mathcal{P}$ and a basis $\bar{B} \subseteq \mathbb{Z}[x_1, \dots, x_r]$ such that $\{I_h\}_{h\in H^p, p>\bar{p}}$ are $\mathcal{F}(\bar{B})$-linked and $G_p = \{1\} \iff G_0 = \{1\}$. Thus it is sufficient to compute the finite set $\mathcal{G} = \{G_0\} \cup \{G_i\}_{2\le i<\bar{p},\ p\in\mathcal{P}}$ to perform our test, in fact:

- for $h \in \mathcal{P}$, $h \ge \bar{p}$: $G_h = \{1\}$ if and only if $G_0 = \{1\}$

- for $h \in \mathcal{P}$, $0 \leq h < \bar{p}$: $G_h \in \mathcal{G}$
- for $h \notin \mathcal{P}$: $G_h = G_p$ for some $p \in \mathcal{P}$ or $G_k = G_0$, for (8.1) and (8.2).

Let us take $\mathcal{G}' = \{G_h \mid G_h \in \mathcal{G},\ G_h \neq \{1\}\}$. We have three cases.

i. $G_h = \{1\}$ for all $G_h \in \mathcal{G}$ (i.e. $\mathcal{G}' = \emptyset$) then $G_h = \{1\},\ \forall\ h \in H$;

ii. $\mathcal{G}' \neq \emptyset$ and there is no $p \in \mathcal{P}$ s.t.

$$G_p \in \mathcal{G}' \quad \text{and} \quad \exists\ h \in H \text{ s.t. } \mathrm{char}(\mathbb{K}_h) = p;$$

in this case the enlarged set $\{I_h\}_{h \in H'}$ has some elements with a non-empty variety, but $\{I_h\}_{h \in H}$ has not.

iii. $\mathcal{G}' \neq \emptyset$ and there is at least a $p \in \mathcal{P}$ s.t.

$$G_p \in \mathcal{G}' \quad \text{and} \quad \exists\ h \in H \text{ s.t. } \mathrm{char}(\mathbb{K}_h) = p;$$

in this case there is an ideal I_h in $\{I_h\}_{h \in H}$ such that $\mathcal{V}(I_h) \neq \emptyset$.

□

We provide an example, which uses the result of Lemma 8.2.12

Example 8.2.13. We take $X = (x, y, z)$, $\{\mathbb{K}_h\}_{h \in H}$ a set of fields. For each $h \in H$ we consider the polynomial ring $\mathbb{K}_h[x, y, z]$ with the lexicograpich order $z < y < x$. Let $B = \{2x + y + z, y^2 + yz + 1\}$ be a basis in $\mathbb{Q}[x, y, z]$ and let $\{I_h\}_{h \in H}$ a set of $\mathcal{F}(B)$-linked ideals with $I_h \in \mathbb{K}_h[x, y, z]$, for each $h \in H$. Observe that B is a reduced Gröbner basis for $I_0 = \mathcal{I}(\mathcal{F}(B), \mathbb{Q})$. If we want to check if there is an $h \in H$ such that $\mathcal{V}(I_h) \neq \emptyset$ we proceed as follows.

- First, we compute $G_0 = G(I_0)$, the reduced Gröbner basis of I_0, recording the maximal coefficients, $\bar{c}$, which appears in the computations. We have:

  $$\begin{aligned}
  &S(2x+y+z, y^2+yz+1) = 2xy^2 + y^3 + y^2z - 2xy^2 - 2xyz - 2x = -2xyz - 2x + y^3 + y^2z\\
  &S(-2xyz - 2x + y^3 + y^2z, 2x+y+z) = -2xyz - 2x + y^3 + y^2z + 2xyz + y^2z + yz^2 = -2x + y^3 + 2y^2z + yz^3\\
  &S(-2x + y^3 + 2y^2z + yz^3, 2x+y+z) = -2x + y^3 + 2y^2z + yz^3 + 2x + y + z = y^3 + 2y^2z + yz^3 + y + z\\
  &S(y^3 + 2y^2z + yz^3 + y + z, y^2+yz+1) = y^3 + 2y^2z + yz^3 + y + z - y^3 - y^2z - y = y^2z + yz^3 + z\\
  &S(y^2z + yz^3 + z, y^2+yz+1) = y^2z + yz^3 + z - y^2z - yz^3 - z = 0.
  \end{aligned}$$

  Then $G_0 = \{2x + y + z, y^2 + yz + 1\}$ and the maximum absolute value of the coefficients which appears in the computation is $\bar{c} = 2$.

- We set $\bar{p}$ as the smallest integer larger than $\bar{c}$, i.e. $\bar{p} = 3$.

- For any prime p less than $\bar{p}$ we compute G_p, the reduced Gröbner basis of $I_p = \mathcal{I}(\mathcal{F}(B), \mathbb{F}_p)$, i.e. $G_2 = \{1\}$.
- We collect G_0, G_2 in $\mathcal{G} = \{G_0, G_2\}$.
- For any $h \in H$ we have that:

  a) if $\mathrm{char}(\mathbb{K}) \geq 3$ than $G_h = G(I_h)$ is G_0,

  b) if $\mathrm{char}(\mathbb{K}) = 2$ than $G_h = G(I_h)$ is G_2.
- Since $G_0 \neq \{1\}$ and $G_2 = \{1\}$, we conclude that for $h \in H$ with $I_h \subseteq \mathbb{K}_h$, if $\mathrm{char}(\mathbb{K}_h) = 2$ then $\mathcal{V}(I_h) = \emptyset$, otherwise $\mathcal{V}(I_h) \neq \emptyset$.

8.3 The maximal root function

In Section 3.1 we have seen how the maximal root function f can be characterized with respect to the distance of cyclic codes. We reformulate the result of Theorem 3.1.18. Let $C \in \mathcal{C}_n$ be a cyclic code of length n over any $\mathbb{F}_q$, with defining set S_C and distance $\mathrm{d}(C)$. For any $(n, S) \in \mathcal{D}$ (see Definition 3.1.1), the maximal root function can be described as follows:

$$\mathsf{f}(n, S) = \min\{\mathrm{d}(C) \mid C \in \mathcal{C}_n,\ S_C = S\}.$$

In principle, from this characterization and the result of Theorem 5.2.1, to compute $\mathsf{f}(n, S)$ for a given pair $(n, S) \in \mathcal{D}$ we have to execute Algorithm A of Section 5.2 for all $q = p^m$, $p \in \mathcal{P}$, $m \geq 1$. This is obviously a non-effective algorithm, since, even if the Algorithm A requires a finite time, we have to perform it for an infinite number of times. We can do slightly better, considering a different system rather than $\hat{J}_C(w)$, which is used in Algorithm A.

Let C be an $[n, k, d]$ cyclic code over $\mathbb{F}_q$ and let w be an integer, $1 \leq w \leq n$. We are now ready to introduce two systems depending on w and C, $\tilde{J}_C(w)$ and $\bar{J}_C(w)$, which are strictly related to the system $\hat{J}_C(w)$ defined in Theorem 5.2.1. First we recall the definition of the polynomials $p_{i,j}$, then we define $\tilde{J}_C(w)$.

Definition 8.3.1. *Let $\mathbb{K}$ be a field. Let $n \geq 2$, $r \geq 1$. Let i, j be two integers s.t. $1 \leq i \neq j \leq r$. We denote by $\mathbf{p_{i,j}}$ the following polynomials in $\mathbb{K}[x_1, \ldots, x_r]$*

$$p_{i,j}(x_i, x_j) = \sum_{h=0}^{n-1} x_i^h x_j^{n-1-h} = \frac{x_i^n - x_j^n}{x_i - x_j}$$

Definition 8.3.2. *Let C be an $[n,k,d]$ cyclic code over $\mathbb{F}_q$ with complete defining set $S_C = \{h_1,\dots h_{n-k}\}$. Let w be an integer such that $1 \leq w \leq n$. Let $p_{i,j} \in \mathbb{F}_q[z_1,\dots,z_w]$ as in Definition 8.3.1. We denote by $\tilde{J}_C(w)$ the following polynomial system in $\mathbb{F}_q[z_1,\dots,z_w,y_1,\dots,y_w,t]$:*

$$\mathbf{\tilde{J}_C(w)} = \begin{cases} y_1 z_1^{h_1} + \dots + y_w z_w^{h_1} = 0 \\ \dots \\ y_1 z_1^{h_{n-k}} + \dots + y_w z_w^{h_{n-k}} = 0 \\ z_1^n - 1 = 0 \\ \dots \\ z_w^n - 1 = 0 \\ y_1^q - y_1 = 0 \\ \dots \\ y_w^q - y_w = 0 \\ p_{1,2}(z_1,z_2) = 0 \\ \dots \\ p_{i,j}(z_i,z_j) = 0 \\ \dots \\ p_{w-1,w}(z_{w-1},z_w) = 0 \\ t y_1 \dots y_w - 1 = 0 \end{cases} \tag{8.3}$$

For any $C \in \mathcal{C}$, system $\tilde{J}_C(w)$ is nothing else that the system obtained by applying the Rabinovich trick to the system $\hat{J}_C(w)$. In particular, the two solution sets are in bijection and hence we may formulate two theorems, which are the analogous of Theorem 5.2.1.

Theorem 8.3.3. *Let C and $[n,k,d]$ cyclic code over $\mathbb{F}_q$. Then, for $1 \leq w \leq n$, $\mathcal{V}$:*

$$A_w(C) \neq 0 \iff \mathcal{V}(\tilde{J}_C(w)) \neq \emptyset \iff G(\tilde{J}_C(w)) \neq \{1\}$$

Theorem 8.3.4. *Let C be a cyclic code over $\mathbb{F}_q$. Then C has distance δ if and only if*

$$\mathcal{V}(\tilde{J}_C(w)) = \emptyset,\ 1 \leq w \leq \delta - 1 \quad \textit{and} \quad \mathcal{V}(\tilde{J}_C(\delta)) \neq \emptyset$$

The other system we introduce, $\bar{J}_C(w)$, is obtained from $\tilde{J}_C(w)$, by removing the equations $y_i^q - y_i$, for $1 \leq i \leq w$.

$$\mathbf{\bar{J}_C(w)} = \begin{cases} y_1 z_1^{h_1} + \cdots + y_w z_w^{h_1} = 0 \\ \dots \\ y_1 z_1^{h_{n-k}} + \cdots + y_w z_w^{h_{n-k}} = 0 \\ z_1^n - 1 = 0 \\ \dots \\ z_w^n - 1 = 0 \\ p_{1,2}(z_1, z_2) = 0 \\ \dots \\ p_{i,j}(z_i, z_j) = 0 \\ \dots \\ p_{w-1,w}(z_{w-1}, z_w) = 0 \\ t y_1 \dots y_w - 1 = 0 \end{cases} \tag{8.4}$$

We need some preliminaries results to understand the gain in using $\bar{J}_C(w)$.

Lemma 8.3.5. *Let $\mathbb{K} = \mathbb{F}_{p^m}$ be a finite field with $p \in \mathcal{P}$, $m \geq 1$. Let I be an ideal in $\mathbb{K}[x_1, \dots, x_r]$. Then*

$$\mathcal{V}(I) \neq \emptyset \iff \exists\ s \geq 1 \textit{ s.t. } \mathcal{V}_{\mathbb{E}}(I) \neq \emptyset, \quad \textit{with } \ \mathbb{E} = \mathbb{F}_{p^{ms}}$$

Proof. If $\mathcal{V}(I) = \emptyset$ then obviously there are not any rationals points in any extension field.

Otherwise, if $\mathcal{V}(I) = \emptyset$, let $(\bar{x}_1, \dots, \bar{x}_r) \in \mathcal{V}(I)$. Any $\bar{x}_i$ must lie in a finite-dimensional extension of $\mathbb{K}$, because $\overline{\mathbb{K}} = \cup_{j=1}^{\infty} \mathbb{F}_{p^{mj}}$. For any $1 \leq i \leq r$, let s_i be an integer s.t. $\bar{x}_i \in \mathbb{F}_{p^{ms_i}}$. We define s and $\mathbb{E}$ as

$$s = \prod_{i=1,\dots,r} s_i, \qquad \mathbb{E} = \mathbb{F}_{p^{ms}}.$$

Then $\bar{x}_i \in \mathbb{F}_{p^{ms_i}} \subseteq \mathbb{E}$ for any $1 \leq i \leq r$ and hence $(\bar{x}_1, \dots, \bar{x}_r) \in \mathcal{V}(I)$, which implies that $\mathcal{V}_{\mathbb{E}}(I) \neq \emptyset$, as required. □

Lemma 8.3.6. *Let w be a fixed integer, $w \geq 1$. Let $p \in \mathcal{P}$ be any prime. Let $\bar{J}_p$ the ideal in $\mathbb{F}_p[z_1, \dots, z_w, y_1, \dots, y_w, t]$ generated by the polynomials of $\bar{J}_C(w)$. Let $\tilde{J}_{p^s}$ the ideal in $\mathbb{F}_{p^s}[z_1, \dots, z_w, y_1, \dots, y_w, t]$ generated by the polynomials of $\tilde{J}_C(w)$. Then*

$$\mathcal{V}(\bar{J}_p) = \emptyset \iff \mathcal{V}(\tilde{J}_{p^s}) = \emptyset, \quad \forall\ s \geq 1, .$$

Proof.

$\Longrightarrow$. Since $\overline{\mathbb{F}}_{p^s} \subseteq \overline{\mathbb{F}}_p$ for any $s \geq 1$, we have $\mathcal{V}(\bar{J}_{p^s}) \subseteq \mathcal{V}(\bar{J}_p)$. On the other hand $\bar{J}_{p^s} \subseteq \tilde{J}_{p^s}$, which implies $\mathcal{V}(\bar{J}_{p^s}) \supseteq \mathcal{V}(\tilde{J}_{p^s})$. Collecting all these inclusions together, we obtain:

$$\mathcal{V}(\tilde{J}_{p^s}) \subseteq \mathcal{V}(\bar{J}_{p^s}) \subseteq \mathcal{V}(\bar{J}_p) = \emptyset$$

which implies $\mathcal{V}(\tilde{J}_{p^s}) = \emptyset$.

$\Longleftarrow$. Let us suppose $\mathcal{V}(\bar{J}_p) \neq \emptyset$ then for Lemma 8.3.5 there exist $s \geq 1$ such that $\mathcal{V}_{\mathbb{F}_{p^s}}(\bar{J}_p) \neq \emptyset$. On the other hand $\mathcal{V}_{\mathbb{F}_{p^s}}(\bar{J}_p) = \mathcal{V}(\bar{\bar{J}}_p)$, where

$$\bar{\bar{J}}_p = \bar{J}_p + \langle z_1^{p^s} - z_1, \ldots, z_w^{p^s} - z_w, y_1^{p^s} - y_1, \ldots, y_w^{p^s} - y_w, t^{p^s} - t \rangle.$$

Since $\tilde{J}_{p^s} \subseteq \bar{\bar{J}}_p$ we obtain $\mathcal{V}(\tilde{J}_{p^s}) \supseteq \mathcal{V}(\bar{\bar{J}}_p) \neq \emptyset$.

□

Lemma 8.3.7. *Let $(n, S) \in \mathcal{D}$. For any $p \in \mathcal{P}_n$, there is an integer $m_p \geq 1$ and a cyclic code C_p over $\mathbb{F}_{p^{m_p}}$ with length n, distance $\mathrm{d}(C_p)$ and complete defining set $S_{C_p} = S$ s.t.*

$$\mathsf{f}(n, S) = \min_{p \in \mathcal{P}_n} \{\mathrm{d}(C_p)\}.$$

Proof. Let $p \in \mathcal{P}$. Let m_p the smallest integer s.t. $m_p \geq 1$ and $n \mid p^{m_p} - 1$, so that $\mathbb{F}_{p^{m_p}}$ is the splitting field of $x^n - 1$ over $\mathbb{F}_p$. Let α be a primitive element of $\mathbb{F}_{p^{m_p}}$. Let g be the polynomial

$$g(x) \in \mathbb{F}_{p^{m_p}}, \quad g(x) = \prod_{i \in S} (x - \alpha^i).$$

Let C_p the cyclic code of length n over $\mathbb{F}_{p^{m_p}}$ generated by g. Then it is obvious that $S_{C_p} = S$ (the cyclotomic cosets are singletons in $\mathbb{F}_{p^{m_p}}$). We have only to show

$$\min\{\mathrm{d}(C) \mid C \in \mathcal{C}_n,\ S_C = S\} = \min_{p \in \mathcal{P}_n} \{\mathrm{d}(C_p)\}.$$

It is enough to show

$$\mathrm{d}(C_p) = \min\{\mathrm{d}(C) \mid C \in \mathcal{C}_n,\ S_C = S,\ \chi(C) = p\}, \tag{8.5}$$

where $\chi(C)$ is as in Definition 3.1.2. In this case:

$$\min_{p \in \mathcal{P}_n} \{\mathrm{d}(C_p)\} = \min_{p \in \mathcal{P}_n} \{\min\{\mathrm{d}(C) \mid C \in \mathcal{C}_n,\ S_C = S,\ \mathrm{char}(C) = p\}\}$$

which is obviously equal to

$$\min\{\mathrm{d}(C) \mid C \in \mathcal{C}_n,\ S_C = S\}.$$

But (8.5) follows immediately from Proposition 3.1.8, since all cyclic codes with the same length, same complete defining set, and same field characteristic have the same distance. □

We denote by $\bar{J}_{p^s}(w)$ the ideal in $\mathbb{F}_{p^s}[z_1,\dots,z_w,y_1,\dots,y_w,t]$ associated to the system $\bar{J}_{C_p}(w)$, for any $s \geq 1$. Similarly, with $\tilde{J}_{p^s}(w)$ we denote the ideal in $\mathbb{F}_{p^s}[z_1,\dots,z_w,y_1,\dots,y_w,t]$ associated to the system $\tilde{J}_{C_p}(w)$.

Lemma 8.3.8. *Let $(n,S) \in \mathcal{D}$ and let p be any prime coprime with n, i.e. $p \in \mathcal{P}_n$. Let m_p be the smallest integer such that $n \mid p^{m_p}-1$ and let C_p be the cyclic code over $\mathbb{F}_{p^{m_p}}$ of length n and complete defining set S. Then $\mathrm{d}(C_p)=d$ if and only if*

$$\mathcal{V}(\bar{J}_p(w))=\emptyset,\ 1\leq w\leq d-1 \quad \text{and} \quad \mathcal{V}(\bar{J}_p(d))\neq\emptyset.$$

Proof.

$\Longleftarrow$ **.** For $1\leq w\leq d-1$ if $\mathcal{V}(\bar{J}_p(w))=\emptyset$ then, from Lemma 8.3.6, $\mathcal{V}(\tilde{J}_{p^s}(w))=\emptyset$ for any $s\geq 1$. In particular, for any s we have that $\mathcal{V}(\tilde{J}_{p^{sm_p}}(w))=\emptyset$, where $1\leq w\leq d-1$.

On the other hand, if $\mathcal{V}(\bar{J}_p(d))\neq\emptyset$ then there is an $s\geq 1$ s.t. $\mathcal{V}(\tilde{J}_{p^s}(d))\neq\emptyset$. In particular, we have $\emptyset\neq\mathcal{V}(\tilde{J}_{p^s}(d))\subseteq\mathcal{V}(\bar{J}_{p^{sm_p}}(d))$, which implies $\mathcal{V}(\tilde{J}_{p^{m_ps}}(d))\neq\emptyset$. We have proved that there is an $s\geq 1$ s.t

$$\mathcal{V}(\tilde{J}_{p^{sm_p}}(w))=\emptyset,\ 1\leq w\leq d-1 \quad \text{and} \quad \mathcal{V}(\bar{J}_{p^{sm_p}}(d))\neq\emptyset.$$

Then from Theorem 8.3.4 there is a code C over $\mathbb{F}_{p^{sm_p}}$, with distance d. But this code has same length, same defining set and same characteristic of C_p, then for Proposition 3.1.8, they have also the same distance.

$\Longrightarrow$ **.** If $\mathrm{d}(C_p)=d$, by Theorem 8.3.4, $\mathcal{V}(\tilde{J}_{p^{m_p}}(d))\neq\emptyset$ and it implies, by Lemma 8.3.6, $\mathcal{V}(\bar{J}_p(d))\neq\emptyset$.

Let us suppose that for some w, $1\leq w\leq d-1$, we have $\mathcal{V}(\bar{J}_p(w))\neq\emptyset$, then $\mathcal{V}(\tilde{J}_{p^s}(w))\neq\emptyset$ for some $s\geq 1$. In particular, for such s it holds $\mathcal{V}(\tilde{J}_{p^{sm_p}}(w))\neq\emptyset$. But then we have a code $C_{p^{sm_p}}$, of length n, defining set S, defined of distance less than d. But this is not possible because for Proposition 3.1.8 $\mathrm{d}(C_{p^{sm_p}})=\mathrm{d}(C_p)=d$. Hence $\mathcal{V}(\bar{J}_p(w))\neq\emptyset$ for any $1\leq w\leq d-1$.

□

Finally, we are ready for the main result of this section.

Theorem 8.3.9. *Let (n,S) be any element of $\mathcal{D}$. Then the value of the optimal root function, $\mathsf{f}(n,S)$, can be computed in a finite time.*

Proof. Let p be any prime coprime with n, i.e. $p\in\mathcal{P}_n$. For any such p, let m_p be the smallest integer such that $n\mid p^{m_p}-1$ and let C_p the cyclic code generated by

$g(x) = \prod_{i\in S}(x - \alpha^i)$, where α is any primitive n-th root of unity over $\mathbb{F}_p$. From Lemma 8.3.7, we have:

$$\mathsf{f}(n,S) = \min_{p\in\mathcal{P}_n}\{\mathrm{d}(C_p)\}, \tag{8.6}$$

where $\mathrm{d}(C_p)$ indicates the distance of C_p. From Lemma 8.3.8, we have that (8.6) becomes

$$\begin{aligned}\mathsf{f}(n,S) &= \min_{p\in\mathcal{P}_n}\{w \mid \mathcal{V}(\bar{J}_p(w)) \neq \emptyset,\ \mathcal{V}(\bar{J}_p(w-1)) = \cdots = \mathcal{V}(\bar{J}_p(1)) = \emptyset\}\\ &= \min_{p\in\mathcal{P}_n}\{w \mid \mathcal{V}(\bar{J}_p(w)) \neq \emptyset\}\end{aligned}$$

Thus, to compute $\mathsf{f}(n,S)$ we have to check the minimum w, $1 \leq w \leq n$, such that there is a prime $p \in \mathcal{P}_n$ with $\mathcal{V}(\bar{J}_p(w)) \neq \emptyset$. But $\{\bar{J}_p(w)\}_{p\in\mathcal{P}_n}$ is a set of F-linked ideals and thanks to Theorem 8.2.12 we can do this check in a finite time for each w. Since the number of w to check is finite, the time needed to compute $\mathsf{f}(n,S)$ is finite. □

Following the proof of Theorem 8.3.9, we propose an algorithm which, given any $(n,S) \in \mathcal{D}$, returns $\mathsf{f}(n,S)$ in a finite time.

Algorithm D

```
Input
A pair $(n,S) \in \mathcal{D}$.
A value $w = 1$.
Output
$\mathsf{f}(n,S)$.
Cycle
Construct the system $\bar{J}_0(w)$ in $\mathbb{Q}[X]$.
Compute $\bar{p}$ as in Lemma 8.2.11.
Compute $\mathcal{G} = \{G(\bar{J}_p(w)\} \cup \{\bar{J}_p(w)\}$ for $p \in \mathcal{P}_n$, $p < \bar{p}$.
If for all $G \in \mathcal{G}$, $G = \{1\}$, then increase $w$ to $w+1$.
Last step
Output $w$.
```

We conclude this section with an example of computation for $\mathsf{f}(n,S)$.

Example 8.3.10. Let (n,S) be a pair in $\mathcal{D}$ with $n = 6$, $S = \{0,1,3\}$. For the BCH bound we have that $\mathsf{f}(n,S) \geq 3$ and we ask if $\mathsf{f}(n,S) = 3$. We consider $X = (z_1, z_2, z_3, y_1, y_2, y_3, t)$ with `DegRevLex` ordering induced by $z_1 > \cdots > z_3 > y_1 \cdots > y_3 > t$. Let $\bar{J}_0(3) = I(\bar{J}_C(3))$ the ideal in $\mathbb{Q}[X]$ associated to the system $\bar{J}_C(3)$. If we compute the $G_0 = G(J_0(3))$ the reduced Gröbner basis of $J_0(3)$, we obtain:

$$G_0 = \{y_1 + y_2 + y_3, z_1 + z_2 + z_3, y_2^2 + y_2y_3 + y_3^2, z_3y_2 - z_2y_3, z_2y_2 + z_2y_3 + z_3y_3, z_2^2 + z_2z_3 + z_3^2y_3^3t - 1z_3^6 - 1z_2z_3^5 - y_2y_3^2t\}.$$

In particular, $G_0 \neq \{1\}$, so there is $p \in \mathcal{P}_n$ such that $\mathcal{V}((\tilde{J}_{p^s}(3)) \neq \emptyset$. Hence, $\mathsf{f}(n, S) = 3$.

Part III

Appendix

9.1 Programs for the root bounds

In this chapter we provide our implementations of the BCH bound, the HT bound, the BS bound, bound I, bound II and bound C. We used these programs to compute Tabular 6.2-6.3-6.4-6.5-6.6-6.7 in Section 6.1.

```
/*
USAGE:   dfset(F,n,g); F a field, n aninteger, g a polynomial
RETURN:  a list of 0 and 1, representin the complete defining
         set of the code over F with length n and generator polynomial g.
         L[i]==0 if in the definig set, L[i]==0, otherwise.
*/

function dfset(F,n,g)
 local R, E,a,Sc,L;
 R<x>:=PolynomialRing(F);
 E<b>:=SplittingField(x^n-1);
 a:=RootOfUnity(n,F);
 Sc:= {i: i in [0..n-1] | Evaluate(g,a^i) eq 0};
 L:=[1: i in [1..n]];
 for i in Sc do
  L[i+1]:=0;
 end for;
 return L;
end function;

/*
USAGE:  Invariant(F,n,g); F a field, n aninteger, g a polynomial
RETURN:  a list containig the complete defining sets of all
codes over F with length n and naturally equivalent
to the code generated by g.
*/

function Invariant(F,n,g,option)
 local R, E,a,Sc,SSc,L, LL;
 R<x>:=PolynomialRing(F);
 E<b>:=SplittingField(x^n-1);
 a:=RootOfUnity(n,F);
 Sc:= {i: i in [0..n-1] | Evaluate(g,a^i) eq 0};
 LL:=[];
 k:=1;
 while (k lt n) do
 if (GCD(k,n) eq 1) then
    SSc:={(k*j) mod n: j in Sc};
    L:=[1: i in [1..n]];
 if (option eq 1) then
  for i in SSc do
   L[i+1]:=0;
  end for;
  LL cat:=[L];
 else
  LL cat:=[SSc];
 end if;
 end if;
 k+:=1;
 end while;
// LL;
 if (#LL eq EulerPhi(n)) then
//controllo che la cardinalità sia giusta e poi tolgo le ripetizioni
  SSc:={J: J in LL};
  LL:=[J: J in SSc];
   return LL;
 else return "error";
 end if;
end function;

/*
USAGE:  AllCyclicCodes(n,F); n an integer, F a field
RETURN: a list containing all the generator polynomials
of cyclic codes of length n and over F, except
for the whole space and the null-code.
*/

function AllCyclicCodes(n,F)
 R<x>:=PolynomialRing(F);
 Fp:=Factorization(x^n-1);
 nf:=#Fp;
 LL:=[];
 for i in [1..2^nf-2] do

  L:=IntegerToSequence(i,2);
  g := 1;
 for j in [1..#L] do
   if L[j] eq 1 then
      g := g*Fp[j][1];
   end if;
 end for;
// Uncomment here to have the list of the codes
// LL cat:=[CyclicCode(n,g)];
//  Uncomment here to have the list of the generator polynomials
   LL cat:=[g];
//  Uncomment here to have the list of the def. sets
// LL cat:=[dfset(F,n,g)]
 end for;
 return LL;
end function;

/*
USAGE:  block(a,b,M); M a list, a, b intengers less than or equal to
the size of M
RETURN: if b>a the list [M[a], M[a+1], ... , M[b]]
        else [M[b], M[b+1], ..., M[1],...,M[a]]
*/

function block(a,b,M)
 if ( a le b) then
return M[a..b];
 else
return M[a..#M] cat M[1..b];
 end if;
end function;

/*
USAGE:   bch(M); M a list of 0 and 1
RETURN:  the bch bound for the codes having M as complete def. set
*/

function bch(M)
 count:= 0;
 bound:= 0;
 z:=#M;
 for i  in [1..z] do
  if (M[i] eq 0) and (bound lt z) then
   count+:=1;
   bound:=Max(bound, count+1);
  else count:=0;
  end if;
  end for;
 if (count ne 0) then
  i:=1;
  while ( M[i] eq 0 ) and (bound lt z) do
   count+:=1;
   bound:=Max(bound, count+1);
   i+:=1;
  end while;
 end if;
 return bound;
end function;

/*
USAGE:   ht(M); M a list of 0 and 1
RETURN:  the ht bound for the codes having M as complete def. set
*/

function ht(M)
n:=#M;  // length of the cyclic code
dist:=bch(M);
lmax:=dist-1;  // max length for the zero block (0^l D^(r-l))
ix:=0;   // starting point for the block
sx:=0; // counter for the blocks
lx:=0; // length of the zero-block
rx:=0; // length of the block (0^l D^(r-l))

for l in [1..lmax] do
 for r in [l..n] do
  gg:= GCD(r,n);
   if (gg le l ) then
     for i in [1..n] do
        bzeri:=i;
s:=0;

while (block(((bzeri-1) mod n +1),((bzeri +l-2) mod n +1),M)
       eq [0:j in [1..l]]) do
// M[((bzeri-1) mod n +1)..((bzeri +l-2) mod n +1)];
        s+:=1;
bzeri:=i+r*s;
end while;
if ((l+s) ge dist) then
dist:=l+s;
lx:=l;
rx:=r;
sx:=s;
ix:=i;
end if;
      end for;
   end if;
 end for;
end for;
```

```
// Uncomment here to see the block which returns ht(M)
/*
printf"inizio: %o  ", ix;
printf"l: %o  ", lx;
printf"s: %o  ", sx;
printf"r: %o  ", rx;
printf"dist: %o  ", dist;
printf"\n";
*/
return dist;
end function;

/*
USAGE:   roos(M); M a list of 0 and 1
RETURN:  the roos bound for the codes having M as complete def. set
*/

function roos(M)
n:=#M;  // length of the cyclic code
dist:=bch(M);
lmax:=dist-1;  // max length for the zero block (0^l D^(r-l))
ix:=0;   // starting point for the block
sx:=0; // counter for the blocks
lx:=0; // length of the zero-block
rx:=0; // length of the block (0^l D^(r-l))
hx:=0;
holes:=0; // counter for the holes

for l in [1..lmax] do
 for r in [1..n] do
  gg:= GCD(r,n);
   if (gg eq 1) then
     for i in [1..n] do
        bzeri:=i;
s:=0;
        holes:=0;

while (holes lt l) do
if (block(((bzeri-1) mod n +1),((bzeri +l-2) mod n +1),M)
    eq [0:j in [1..l]]) then
 s+:=1; // found a block
else
 holes+:=1; // found a hole
end if;
bzeri:=i+r*(s+holes);
```

```
end while;
if ((l+s) ge dist) then
dist:=l+s;
lx:=l;
rx:=r;
sx:=s;
ix:=i;
  hx:=holes;
end if;
      end for;
   end if;
 end for;
end for;

// Uncomment here to see the block which returns roos(M)
/*
printf"inizio: %o  ", ix;
printf"l: %o  ", lx;
printf"s: %o  ", sx;
printf"r: %o  ", rx;
printf"holes: %o", hx;
printf"dist: %o  ", dist;
printf"\n";
*/
return dist;
end function;

/*
USAGE:   bs(M); M a list of 0 and 1
RETURN:  the "straight-version" of Betti-Sala bound
 for the codes having M as complete def. set
*/

function bs(M);
n:=#M;
d:=bch(M);
lmax:=d-1;
ix:=0;
lx:=0;
mx:=0;

for l in [1..lmax] do // l is the length of the blocks
 mMax:=Floor(lmax/l);
 for m in [1..mMax] do  // m is the number of blocks
  for i in [1..n] do
```

```
   if (block(i,((i+l*m-2) mod n +1),M)  // found a long block
       eq [0: j in [1..(m*l)]]) then
s:=0;
while ( ( block(((i+l*m+s*l) mod n+1),((i+ l*m+(s+1)*l-2) mod n +1),M)
         eq [0: j in [1..(l-1)]] )  and (s le m)  )do
s+:=1;
end while;
if s le m then  // the small blocks are not enough
continue i;
else  // small blocks found
if (m*l+l gt d) then
d:=m*l+l;
ix:=i;
lx:=l;
mx:=m;
break i;
end if;
end if;
end if;

end for;
end for;
end for;

// Uncomment here to see the pattern which returns bs(M)
/*
printf"inizio: %o  ", ix;
printf"l: %o  ", lx;
printf"m: %o  ", sx;
printf"\n";
*/

return d;
end function;

/*
USAGE:   BS(M); M a list of 0 and 1
RETURN:  the Betti-Sala bound for the codes having M as complete def. set
*/

function BS(M)
 return Max(bs(M), bs(Reverse(M)));
end function;

/*
USAGE:   b2(M); M a list of 0 and 1
```

```
RETURN:  the straight-version of bound II (Prop. 7.0.8)
 for the codes having M as complete def. set
*/

function b2(M);
n:=#M;
d:=bch(M);
lmax:=d-1;
ix:=0;
lx:=0;
mx:=0;
sx:=0;

for l in [1..lmax] do // l is the length of the zero-blocks
 mMax:=Floor(lmax/l);
  if (n mod l) ne 0 then
 for m in [1..mMax] do  // m is the number of blocks
  for i in [1..n] do

   if (block(i,((i+l*m-2) mod n +1),M) eq [0: j in [1..(m*l)]]) then
s:=0;

while ( block(((i+l*m+s*l) mod n+1),((i+ l*m+(s+1)*l-2) mod n +1),M)
       eq [0: j in [1..(l-1)]] )do
s+:=1;
end while;

if s le m then  // the small blocks are not enough
continue i;
else  // small blocks found
if (m*l+l +s-m-1 ge d) then
d:=m*l+l+s-m-1;
ix:=i;
lx:=l;
mx:=m;
sx:=s;
end if;
end if;
end if;

end for;
end for;
   end if;
end for;

//Uncomment here to see the pattern which returns b2(M)
```

```
/*
printf"inizio: %o  ", ix;
printf"l: %o  ", lx;
printf"m: %o  ", mx;
printf"s: %o ", sx;
printf" dist: %o ",d;
printf"\n";
*/

return d;
end function;

/*
USAGE:   b1(M); M a list of 0 and 1
RETURN:  the straight-version of bound I (Prop. 7.0.5)
 for the codes having M as complete def. set
*/

function b1(M)
n:=#M;  // length of the code
dist:=bch(M);
lmax:=dist-1;  // max length of the zero-block (0^l D^r)(0^m D^r)^s
ix:=0;   // staring point of the block
sx:=0; // counter for the small blocks
lx:=0; // length of the long block
mx:=0; // length of the small zero-block
rx:=0; // length of the small delta-block
dx:=0;

for l in [1..lmax] do
for m in [1..l] do
   for r in [1..n-m-1] do
  gg:= GCD(m+r,n);
   if (gg le m ) then
     for i in [1..n] do
if (block(i,((i+l-2) mod n +1),M) eq [0: j in [1..l]]) then
  // long block found
bzeri:=i+l+r;
s:=0;

while (block(((bzeri-1) mod n +1),((bzeri +m-2) mod n +1),M)
       eq [0:j in [1..m]]) do
s+:=1;
bzeri+:=(m+r);
end while;
```

```
dx:= l+s+1-Floor(l/(m+r))*r-Max(0,((l mod (m+r)) -m) );
if (dx ge dist) then
dist:=dx;
lx:=l;
mx:=m;
rx:=r;
sx:=s;
ix:=i;
end if;
end if;
      end for;
   end if;
   end for;
end for;
end for;

//Uncomment here to see the pattern which returns b1(M)
/*
printf"inizio: %o  ", ix;
printf"l: %o  ", lx;
printf"m: %o  ", mx;
printf"s: %o  ", sx;
printf"r: %o  ", rx;
printf"dist: %o  ", dist;
printf"\n";
*/
return dist;
end function;

/*
USAGE:   B1(M); M a list of 0 and 1
RETURN:  the bound I (Proposition 7.05-7.06)
 for the codes having M as complete def. set
*/

function B1(M)
 return Max(b1(M), b1(Reverse(M)));
end function;

/*
USAGE:   B2(M); M a list of 0 and 1
RETURN:  the bound II (Proposition 7.08-7.09)
 for the codes having M as complete def. set
*/

function B2(M)
```

```
 return Max(b2(M), b2(Reverse(M)));
end function;

/*
USAGE:   bC(M); M a list of 0 and 1
RETURN:  the bound C (Theorem 7.1.13)
 for the codes having M as complete def. set
*/

function bC(M)
 return Max(B2(M),B1(M));
end function;

/*
USAGE:   bC2(M); M a list of 0 and 1
RETURN:  the maximum between bC and BS
 for the codes having M as complete def. set
*/

function bC2(M)
 return Max(bC(M),BS(M));
end function;

/*
USAGE:   testTight(Field, n, filename); Field a field,
 n an integer, filename a string
RETURN:  - a file "filename".out with the number of codes of length
         n for which the implemented bounds are tight
         - a file "filename"_time.out with the times needed for
         the computation
*/

procedure testTight(Field, n, filename)
local F,q;
f2:=filename cat"_time";
q:=#Field;
if GCD(n,q) eq 1 then
  L:=AllCyclicCodes(n, Field);
//  t:=Cputime();
  distL:=[MinimumDistance(CyclicCode(n,j)):j in L];
//  Cputime(t);
  dfsetL:=[dfset(Field,n,j):j in L];
  t:=Cputime();
  BCH:=[bch(M): M in dfsetL];
  time1:=Cputime(t); t:=Cputime();
  HT:=[ht(M): M in dfsetL];
  time2:=Cputime(t); t:=Cputime();
  BetSal:=[BS(M): M in dfsetL];
  time3:=Cputime(t); t:=Cputime();
  ROOS:=[roos(M): M in dfsetL];
  time4:=Cputime(t); t:=Cputime();
  boundC:=[bC(M): M in dfsetL];
  time5:=Cputime(t); t:=Cputime();
  boundC2:=[bC2(M): M in dfsetL];
  time6:=Cputime(t);
  nbch:=0; nbs:=0; nht:=0; nroos:=0; nC:=0; nC2:=0;
   for i in [1..#dfsetL] do
     if (Max({distL[i],BCH[i],BetSal[i],HT[i],ROOS[i],boundC[i],boundC2[i]})
         eq distL[i]) then
        if (distL[i] eq BCH[i]) then
         nbch+:=1;
        end if;
if (distL[i] eq HT[i]) then
         nht+:=1;
        end if;
if (distL[i] eq BetSal[i]) then
         nbs+:=1;
        end if;
if (distL[i] eq ROOS[i]) then
         nroos+:=1;
        end if;
if (distL[i] eq boundC[i]) then
         nC+:=1;
        end if;
if (distL[i] eq boundC2[i]) then
         nC2+:=1;
        end if;
     else
        fprintf F, "\n ERROR ERROR ERROR \n";
        printf "\n ERROR ERROR ERROR \n";
     end if;
   end for;
   fprintf filename, "%5o & %5o & %8o & %8o & %8o & %8o & %8o & %8o \\\\ \n",
      n, #dfsetL, nbch, nht, nbs, nroos, nC,nC2;
   fprintf f2, "%5o &    %5o & %8o & %8o & %8o & %8o & %8o & %8o \\\\ \n",
      n, #dfsetL, time1, time2, time3, time4, time5, time6;
end if;
end procedure;

/*
USAGE:   tightness(n1,n2,Field,filename); Field a field,
 n1, n2, integers, filename a string
```

```
RETURN:  - a file "filename".out with a tabular containing
           the number of codes of length n, n1<= n <= n2
           for which the implemented bounds are tight
         - a file "filename"_time.out with the times needed for
         the computation
*/

procedure tightness(n1,n2,Field,filename)
f2:=filename cat"_time";
fprintf filename,"\\begin{tabular}{c|c|c|c|c|c|c|} \n";
fprintf filename,"\\hline \n";
fprintf filename, " n & N. codes & BCH & HT & BS & ROOS & BC & BC2 \\\\ \n ";
fprintf f2,       " n & N. codes & BCH & HT & BS & ROOS & BC & BC2 \\\\ \n ";
 for i in [n1..n2] do
    if (i-n1) mod 20 eq 0 then
     printf "n= %4o ---> %4o \n", i , n2;
    elif (i-n1) mod 10 eq 0 then
     printf "        ---> \n";
    end if;
    testTight(Field, i, filename);
 end for;
fprintf filename,"\\hline \n";
fprintf filename,"\\end{tabular} \n";
end procedure;
```

9.2 Programs for the strict bounds

In this section we provide our implementations of first and second realization of singleton procedure, which correspond, as described in Section 4.2.

```
/*
USAGE:   CirculantMatrix(v); v a list
RETURN:  the circulant matrix obtained from v
*/

 function CirculantMatrix(v)
local F,n,L;
F:=Parent(v[1]);
L:=[];
n:=#v;
for i in [0..n-1] do
 L cat:= Rotate(v,i);
end for;
return(Matrix(F,n,n,L));
end function;
```

```
/*
USAGE:   Aset(v); v a list of 0 and 1
RETURN:  the A-set of v, (see Def. 2.2.15)
*/

function Aset(v)
local R,r,i,j,L,tmp,vv;
 R:=[i: i in [1..#v] | v[i] eq 1];
 r:=#R;
 vv:=[];L:=[];
  for i in [1..2^r-1] do
   tmp:=IntegerToSequence(i,2);
    if (#tmp lt r) then
     tmp := tmp cat [0:i in [1..(r-#tmp)]];
    end if;
    for j in [1..#v] do
     if j notin R then
      vv[j]:=v[j];
     else
      for k in [1..r] do
       vv[R[k]]:=tmp[k];
      end for;
     end if;
   end for;
  L cat:= [vv];
 end for;
return L;
end function;

/*
USAGE:   Equiv(M); M a list of 0 and 1, option an integer
RETURN:  if option==1 then it returns the list of def. sets
         naturally equivalent to M
         otherwise it returns a list of list with 0 and 1
         representing the def. sets naturally equivalent to M
*/

function Equiv(M,option)
 n:=#M;
 L1:=[];
 L2:=[];
 DS:=[i-1: i in [1..n] | M[i] eq 0];
  for i in [1..n] do
   if GCD(i,n) eq 1 then
    LD:=[(i*j) mod n : j in [0..n] | j in DS];
    LLD:=[1: i in [1..n]];
    for i in LD do
     LLD[i+1]:=0;
    end for;
    L2 cat:=[LLD];
    L1 cat:=[LD];
   end if;
  end for;
if option eq 1 then
 return L1;
else
 return L2;
end if;
end function;

/*
USAGE:   CheckSingleton(M); M a matrix with entries 0 and 1
RETURN:  a list with entries which correspond to a singleton
         in M
*/

function CheckSingleton(M)
n:=Ncols(M);
m:=Nrows(M);
Sing:=[];
 for j in [1..n] do
  w:=[i: i in [1..m] |M[i,j] eq 1];
  if (#w eq 1) then
    Sing cat:=[[i,j]: i in w];
  end if;
 end for;
return Sing;
end function;

/*
USAGE:   RandomSingletonProcedure(M); M a matrix with entries 0 and 1
RETURN:  the number of steps for which the singleton procedure has
         success on M
NOTE:    it is a different implementation of the singleton procedure
         w.r.t. SingletonProcedure, which follows.
*/

function RandomSingletonProcedure(M)
 r:=1;
 MM:=M;
 S:=CheckSingleton(MM);
 r, ")    ", M ;
 while (S ne [] ) and (Nrows(MM) gt 1) do
 n:=Ncols(MM);
 m:=Nrows(MM);
  r+:=1;
  z:=Random(S);
  I:=[i: i in [1..m] | i ne z[1]];
  J:=[j: j in [1..n] | j ne z[2]];
  MM:=Submatrix(MM,I,J);
  r, ")    ", MM , " singoletto: ", z;
  S:= CheckSingleton(MM);
 end while;
return r;
end function;

/*
USAGE:   RandomSingletonProcedure(M); M a matrix with entries 0 and 1
RETURN:  the number of steps for which the singleton procedure has
         success on M
NOTE:    it is a different implementation of the singleton procedure
         w.r.t. RandomSingletonProcedure.
*/

function SingletonProcedure(M)
n:=Ncols(M);
m:=Nrows(M);
S:=CheckSingleton(M);
 if (m eq 1) or (S eq []) then
  return 1;
 elif (S ne []) then
   z:=Random(S);
   I:=[i: i in [1..m] | i ne z[1]];
   J:=[j: j in [1..n] | j ne z[2]];
   M:=Submatrix(M,I,J);
   return 1+SingletonProcedure(M);
 end if;
end function;

/*
USAGE:   Schaub(v); v a list of 0 and 1
RETURN:  r1, the output of the first realization of
         the singleton procedure on M(v), the
         circulat matrix of v
*/

function Schaub(v);
  r:=0; i:=0; L:=[];
  S:=[Rotate(v,i): i in [1..#v]];
  for j in [1..#v] do
   L cat:=[S[j]];
   M:=Matrix(GF(2),#L, #v,L);
   j, " )", M;
   "";
   r:=SingletonProcedure(M);
   r;
   "";
   if (r ne #L) then
     Prune(~L);
   else
     r1:=r;
   end if;
  end for;
  return r1;
end function;

/*
USAGE:   VLint(v); v a list of 0 and 1
RETURN:  r1, the output of the second realization of
         the singleton procedure on M(v), the
         circulat matrix of v.
NOTE:    in this version we use SingletonProcedure
         function to perform the singleton procedure
*/

function VLint(v);
 r:=0;i:=0;
 S:={Rotate(v,i): i in [1..#v]};
 subS:=Subsets(S);
 n:=#subS;
 for j in subS do
  i+:=1;
  if (Floor(i/n*100) mod 10) eq 0 then
   "Progress: -------> ", Floor(i/n*100), " % ";
  end if;
   M:=Matrix(GF(2),#j, #v,[k: k in j]);
   r1:=SingletonProcedure(M);
   if r1 gt r then
     r:=r1;
   end if;
 end for;
 return(r);
end function;
```

```
/*
USAGE:   VLint(v); v a list of 0 and 1
RETURN:  r1, the output of the second realization of
         the singleton procedure on M(v), the
         circulat matrix of v
NOTE:    in this version we use RandomSingletonProcedure
         function to perform the singleton procedure
*/

function VLint2(v);
 r:=0;i:=0;
 S:={Rotate(v,i): i in [1..#v]};
 subS:=Subsets(S);
 n:=#subS;
 for j in subS do
  i+:=1;
  if (Floor(i/n*100) mod 10) eq 0 then
   "Progress: -------> ", Floor(i/n*100), " % ";
  end if;
   M:=Matrix(GF(2),#j, #v,[k: k in j]);
   r1:=RandomSingletonProcedure(M);
   if r1 gt r then
     r:=r1;
   end if;
 end for;
 return(r);
end function;
```

9.3 Computational proofs and numerical confirmations

Some of the examples provided in the thesis have been found computationally. In this section we report the MAGMA ([MAG]) commands we used.

- in Theorem 3.1.20 we provide two codes C_1 over $\mathbb{F}_3$ and C_2 over $\mathbb{F}_{17}$, of length 16 which have complete defining set $S = \{1, 2, 3, 4, 6, 9, 11, 12\}$, claiming that $\mathrm{d}(C_1) = 5$ and $\mathrm{d}(C_2) = 6$. The following MAGMA instructions have been used to prove our claim.

  ```
  > n:=16;
  > S1:={(1*3^i) mod n: i in [1..40]};
  > S1;
  { 1, 3, 9, 11 }
  > S2:={(2*3^i) mod n: i in [1..40]};
  > S2;
  { 2, 6 }
  > S4:={(4*3^i) mod n: i in [1..40]};
  ```

```
> S4;
{ 4, 12 }
> S5:={(5*3^i) mod n: i in [1..40]};
> S5;
{ 5, 7, 13, 15 }
> S8:={(8*3^i) mod n: i in [1..40]};
> S8;
{ 8 }
> S10:={(10*3^i) mod n: i in [1..40]};
> S10;
{ 10, 14 }
> S:=S1 join S2 join S4;
> S;
{ 1, 2, 3, 4, 6, 9, 11, 12 }
>
> R1<x>:=PolynomialRing(GF(3));
> KK<a>:=SplittingField(x^n-1);
> b1:=RootOfUnity(n,GF(3));
> RR1<y>:=PolynomialRing(KK);
> g1y:=1;
> for i in S do
for> g1y:=g1y*(y-b1^i);
for> end for;
> g1x:=R1!g1y;
> C1:=CyclicCode(n,g1x);
> C1;
[16, 8, 5] Cyclic Linear Code over GF(3)
Generator matrix:
[1 0 0 0 0 0 0 0 1 1 1 2 1 0 2 2]
[0 1 0 0 0 0 0 0 1 2 2 0 0 1 2 1]
[0 0 1 0 0 0 0 0 2 0 1 0 2 0 2 0]
[0 0 0 1 0 0 0 0 0 2 0 1 0 2 0 2]
[0 0 0 0 1 0 0 0 1 1 0 2 2 0 1 2]
[0 0 0 0 0 1 0 0 1 2 2 2 0 2 2 0]
[0 0 0 0 0 0 1 0 0 1 2 2 2 0 2 2]
[0 0 0 0 0 0 0 1 1 1 2 1 0 2 2 1]
>
> R2<x>:=PolynomialRing(GF(17));
> KK<a>:=SplittingField(x^n-1);
> b2:=RootOfUnity(n,GF(17));
> RR2<y>:=PolynomialRing(KK);
> g2y:=1;
> for i in S do
for> g2y:=g2y*(y-b2^i);
for> end for;
> g2y;
```

```
y^8 + 10*y^7 + 10*y^6 + 8*y^5 + 15*y^4 + 5*y^3 + 7*y^2 + 7*y + 1
> g2x:=R2!g2y;
> C2:=CyclicCode(n,g2x);
> C2;
[16, 8] Cyclic Linear Code over GF(17)
Generator matrix:
[ 1  0  0  0  0  0  0  0  1  7  7  5 15  8 10 10]
[ 0  1  0  0  0  0  0  0  7 16  5  8  8  3 10 12]
[ 0  0  1  0  0  0  0  0  5  8  0 13 15 14  2  9]
[ 0  0  0  1  0  0  0  0  8 10 13  6 14 11  9 14]
[ 0  0  0  0  1  0  0  0  3 12 14 11  0  4  7  5]
[ 0  0  0  0  0  1  0  0 12  2 11  6  4 11  5  8]
[ 0  0  0  0  0  0  1  0  9  7 14  5  5  8 16 10]
[ 0  0  0  0  0  0  0  1  7  7  5 15  8 10 10  1]
> MinimumDistance(C2);
6
```

- In Theorem 3.5.7 we proved that the Roos bound and the Boston bound V are not strict root bound, claiming that they cannot be proved using singleton procedure. The following instructions provide a computational evidence of our claim.

```
> vBoston:=[0,0,1,0,0,1,0,0,1,1,1,1,1];
> VLint(vBoston);
5
>
>vRoos:=[0,0,1,1,1,1,0,0,1,0,0,1,1,1,1,1,1,1,1,1]
>roos(vRoos);
5
>VLint(vRoos);
4
```

- In Theorem 3.6.3, we provide a code with defining set $S := \{0, 1, 2, 4, 5, 8, 10\}$ which has distance 4 to contradict Theorem 3.6.1. This code has been generated by the following instructions.

```
> R<x>:=PolynomialRing(GF(2));
> b:=RootOfUnity(15,GF(2));
> S:={0,1,2,4,5,8,10};
> g:=1;
> KK:=SplittingField(x^15-1);
> RR<y>:=PolynomialRing(KK);
> for i in S do
for> g:=g*(y-b^i);
for> end for;
> g;
```

```
y^7 + y^3 + y + 1
> gr:=R!g;
> gr;
x^7 + x^3 + x + 1
> C:=CyclicCode(15,gr);
> C;
[15, 8, 4] Cyclic Linear Code over GF(2)
Generator matrix:
[1 0 0 0 0 0 0 0 1 1 0 1 0 0 0]
[0 1 0 0 0 0 0 0 0 1 1 0 1 0 0]
[0 0 1 0 0 0 0 0 0 0 1 1 0 1 0]
[0 0 0 1 0 0 0 0 0 0 0 1 1 0 1]
[0 0 0 0 1 0 0 0 1 1 0 1 1 1 0]
[0 0 0 0 0 1 0 0 0 1 1 0 1 1 1]
[0 0 0 0 0 0 1 0 1 1 1 0 0 1 1]
[0 0 0 0 0 0 0 1 1 0 1 0 0 0 1]
```

- In Theorem 3.6.4, we provide a code with defining set

$$S := \{0, 1, 3, 4, 5, 6, 7, 9, 11, 13, 15, 17, 19\}$$

which has distance 6 to contradict Theorem 3.6.2. This code has been generated by the following instructions.

```
> R<x>:=PolynomialRing(GF(11));
> b:=RootOfUnity(20,GF(11));
> S:={0, 1, 3, 4, 5, 6, 7, 9, 11, 13, 15, 17, 19};
> g:=1;
> KK:=SplittingField(x^20-1);
> RR<y>:=PolynomialRing(KK);
> for i in S do
for> g:=g*(y-b^i);
for> end for;
> g;
y^13 + 9*y^12 + y^10 + y^3 + 9*y^2 + 1
> gr:=R!g;
> gr;
x^13 + 9*x^12 + x^10 + x^3 + 9*x^2 + 1
> C:=CyclicCode(20,gr);
> C;
[20, 7, 6] Cyclic Linear Code over GF(11)
Generator matrix:
[ 1  0  0  0  0  0  0  1  0  9  1  0  0  0  0  0  0  1  0  9]
[ 0  1  0  0  0  0  0  2  1  7  0  1  0  0  0  0  0  2  1  7]
[ 0  0  1  0  0  0  0  4  2  4  0  0  1  0  0  0  0  4  2  4]
[ 0  0  0  1  0  0  0  7  4 10  0  0  0  1  0  0  0  7  4 10]
[ 0  0  0  0  1  0  0  1  7  2  0  0  0  0  1  0  0  1  7  2]
[ 0  0  0  0  0  1  0  9  1  0  0  0  0  0  0  1  0  9  1  0]
[ 0  0  0  0  0  0
```

- in Theorem 4.1.19 we provide two codes C_1 over $\mathbb{F}_{3^5}$ and C_2 over $\mathbb{F}_{2^{10}}$ of length 11 which have complete defining set

  $$S = \{0, 1, 2, 3, 5\}$$

  claiming that $\mathrm{d}(C_1) = 5$ and $\mathrm{d}(C_2) = 6$. The following instructions have been used to prove our claim.

```
> M:=[0,0,0,0,1,0,1,1,1,1,1];
> p:=2;
> F<x>:=PolynomialRing(GF(p));
> K<b>:=SplittingField(x^11-1);
> a:=RootOfUnity(11,GF(p));
> R<y>:=PolynomialRing(K);
> g:=(y-a^0)*(y-a^1)*(y-a^2)*(y-a^3)*(y-a^5);
> C:=CyclicCode(11,g);
> d:=MinimumDistance(C);
> d;
6
>
> p:=3;
> F<x>:=PolynomialRing(GF(p));
> K<b>:=SplittingField(x^11-1);
> a:=RootOfUnity(11,GF(p));
> R<y>:=PolynomialRing(K);
> g:=(y-a^0)*(y-a^1)*(y-a^2)*(y-a^3)*(y-a^5);
> C:=CyclicCode(11,g);
> d:=MinimumDistance(C);
> d;
5
```

- in Remark 4.2.4 we claim that for $\mathbf{v} = (0, 0, 0, \Delta^{+}, 0, \Delta^{+}, 0, 0, \Delta^{+}, \Delta^{+}, \Delta^{+})$ the first rank-bounding algorithm applied to $M(\mathbf{v})$ returns 5 and the second returns 6. We checked this claim with the following instructions.

```
> v:=[0,0,0,1,0,1,0,0,1,1,1];
> VLint(v);
6
> Schaub(v);
5
```

Bibliography

[ABO09] D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, Gröbner Bases, Coding, and Cryptography, Springer, 2009, pp. 47–68.

[ACS90] D. Augot, P. Charpin, and N. Sendrier, *The minimum distance of some binary codes via the newton's identites*, Eurocode '90, LNCS, vol. 514, Springer, 1990, pp. 65–73.

[ACS92] ______, *Studying the locator polynomials of minimum weight codewords of bch codes*, Information Theory, IEEE Transactions on **38** (1992), no. 3, 960–973.

[AL69] H. L. Althaus and R. J. Leake, *Inverse of a finite-field vandermonde matrix*, IEEE Trans. on Inf. Th **15** (1969), 172.

[Aug96] D. Augot, *Description of the minimum weight codewords of cyclic codes by algebraic system*, Finite Fields Appl. (1996), no. 2, 138–152.

[Bet05] Emanuele Betti, *Un'interpretazione algebrica della distanza dei codici ciclici*, Master's thesis (laurea), University of Pisa, Department of Mathematics, 2005.

[Bla83] R. E. Blahut, *Theory and practice of error control codes*, Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1983.

[BM09] M. Byrne and T. Mora, *Gröbner bases over commutative rings and applications to coding theory*, Gröbner Bases, Coding, and Cryptography (M. Sala, T. Mora, L. Perret, S. Sakata, and C. Traverso, eds.), RISC Book Series, Springer, Heidelberg, 2009, pp. 239–261.

[BMvT78] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. on Inf. Th. **24** (1978), no. 3, 384–386.

[Bos01] N. Boston, *Bounding minimum distances of cyclic codes using algebraic geometry*, International Workshop on Coding and Cryptography (Paris, 2001), Electron. Notes Discrete Math., vol. 6, Elsevier, Amsterdam, 2001, p. 10.

[BPW$^+$10] S. Bulygin, R. Pellikaan, I. Woungang, S. Misra, SC. Misra, et al., *Decoding and finding the minimum distance with groebner bases: history and new insights*, Information and Coding Theory (2010).

[BRC60] R. C. Bose and D. K. Ray-Chaudhuri, *On a class of error correcting binary group codes*, Information and Control **3** (1960), 68–79.

[BS05] E. Betti and M. Sala, *A bound for the distance of cyclic codes which is sometimes stronger than the roos bound*, BCRI preprint, www.bcri.ucc.ie, 7, University College Cork, Boole Centre BCRI, UCC Cork, Ireland, 2005.

[BS06] ______, *A new bound for the minimum distance of a cyclic code from its defining set*, IEEE Trans. on Inf. Th. **52** (2006), no. 8, 3700–3706.

[BS07] ______, *A theory for distance bounding cyclic codes*, BCRI preprint, www.bcri.ucc.ie 63, University College Cork, Boole Centre BCRI, UCC Cork, Ireland, 2007.

[Buc65] Bruno Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.

[Buc98] B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.

[Buc06] ______, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), no. 3-4, 475–511.

[Cha98] P. Charpin, *Open problems on cyclic codes*, Handbook of coding theory, Vol. I, II, North-Holland, Amsterdam, 1998, pp. 963–1063.

[Chi72] R. T. Chien, *A new proof of the BCH bound*, IEEE Trans. on Inf. Th. **IT-18** (1972), 541.

[CLO07] D. Cox, J. Little, and D. O'Shea, *Ideals, varieties, and algorithms*, third ed., Springer, 2007, An introduction to computational algebraic geometry and commutative algebra.

[CMSvS91] G. Castagnoli, J. L. Massey, P. A. Schoeller, and N. von Seeman, *On repeated-root cyclic codes*, IEEE Trans. on Inf.. Th. **37** (1991), 337–342.

[Coo90] A. B. III Cooper, *Direct solution of BCH decoding equations*, Comm., Cont. and Sign. Proc. (1990), 281–286.

[Coo91] ______, *Finding BCH error locator polynomials in one step*, Electronic Letters **27** (1991), no. 22, 2090–2091.

[Coo93] ______, *Toward a new method of decoding algebraic codes using Gröbner bases*, Transactions of the Tenth Army Conference on Applied Mathematics and Computing (1992), vol. 93, U.S. Army, 1993, pp. 1–11.

[CRHT94a] X. Chen, I. S. Reed, T. Helleseth, and K. Truong, *General principles for the algebraic decoding of cyclic codes*, IEEE Trans. on Inf. Th. **40** (1994), 1661–1663.

[CRHT94b] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong, *Algebraic decoding of cyclic codes: a polynomial ideal point of view*, Finite fields, Contemp. Math., vol. 168, Amer. Math. Soc., 1994, pp. 15–22.

[CRHT94c] X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong, *Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance*, IEEE Trans. on Inf. Th. **40** (1994), no. 5, 1654–1661.

[CS84] D. Coppersmith and G. Seroussi, *On the minimum distance of some quadratic residue codes*, IEEE Trans. on Inf. Th. **30** (1984), no. 2, part 2, 407–411.

[CU57] L. Carlitz and S. Uchiyama, *Bounds for exponential sums*, Duke Math. J. **24** (1957), 37–41.

[Cur10] M. Curto, *Border bound: un metodo per stimare distanze dei codici ciclici*, Bachelor's thesis (laurea triennale), University of Trento, Department of Mathematics, 2010.

[FT89] G. L. Feng and K. K. Tzeng, *A generalized Euclidean algorithm for multisequence shift-register synthesis*, IEEE Trans. on Inf. Th. **35** (1989), no. 3, 584–594.

[FT91a] ______, *Decoding cyclic and BCH codes up to actual minimum distance using nonrecurrent syndrome dependence relations*, IEEE Trans. on Inf. Th. **37** (1991), no. 6, 1716–1723.

[FT91b] ______, *A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes*, IEEE Trans. on Inf. Th. **37** (1991), no. 5, 1274–1287.

[Hoc59] A. Hocquenghem, *Codes correcteurs d'erreurs*, Chiffres **2** (1959), 147–156.

[HP03] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*, Cambridge University Press, 2003.

[HT72] C. R. P. Hartmann and K. K. Tzeng, *Generalizations of the* BCH *bound*, Information and Control **20** (1972), 489–498.

[KRK88] A. Kandri-Rody and D. Kapur, *Computing a Gröbner basis of a polynomial ideal over a Euclidean domain*, J. Symbolic Comput. **6** (1988), no. 1, 37–57.

[Lev95] F. Levy-dit-Vehel, *Bounds on the minimum distance of the duals of extended BCH codes over* $\mathbf{F}_p$, Appl. Algebra Engrg. Comm. Comput. **6** (1995), no. 3, 175–190.

[MAG] *MAGMA: Computational Algebra System for Algebra, Number Theory and Geometry*, The University of Sydney Computational Algebra Group., http://magma.maths.usyd.edu.au/magma.

[MAI97] T. Matsuo, Y. Araki, and K. Imamura, *Relations between several minimum distance bounds of binary cyclic*, Trans. fundamentals IEICE **E80-A** (1997), 2253–2255.

[MK93] C. J. Moreno and P. V. Kumar, *Minimum distance bounds for cyclic codes and deligne's theorem*, IEEE Trans. on Inf. Th. **39** (1993), 1524–1534.

[MM92] C. J. Moreno and O. Moreno, *An improved Bombieri-Weil bound and applications to coding theory*, J. Number Theory **42** (1992), 32–46.

[MO09] T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, Gröbner Bases, Coding, and Cryptography (M. Sala, T. Mora, L. Perret, S. Sakata, and C. Traverso, eds.), RISC Book Series, Springer, Heidelberg, 2009, pp. 69–91.

[Möl88] H. M. Möller, *On the construction of Gröbner bases using syzygies*, J. Symbolic Comput. **6** (1988), no. 2-3, 345–359.

[Mor05] T. Mora, *Solving polynomial equation systems ii: Macaulay's paradigm and gröbner technology*, vol. 2, Cambridge University Press, 2005.

[Morar] ———, *Solving polynomial equation systems. III, algebraic solving and beyond*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, to appear.

[MS77] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes. I*, North-Holland Publishing Co., Amsterdam, 1977, North-Holland Mathematical Library, Vol. 16.

[MS81] F.J. MacWilliams and N.J.A. Sloane, *The theory of error correcting codes*, NHML016, NH, 1981.

[MS88] J. L. Massey and T. Schaub, *Linear complexity in coding theory*, Coding theory and applications (Cachan, 1986), LNCS, vol. 311, Springer, Berlin, 1988, pp. 19–32.

[MS03] T. Mora and M. Sala, *On the Gröbner bases of some symmetric systems and their application to coding theory*, J. Symbolic Comput. **35** (2003), no. 2, 177–194.

[Pan89] L. Pan, *On the D-bases of polynomial ideals over principal ideal domains*, J. Symbolic Comput. **7** (1989), no. 1, 55–69.

[PHB98] V. S. Pless, W. C. Huffman, and R. A. Brualdi (eds.), *Handbook of Coding Theory. Vol. I, II*, North-Holland, Amsterdam, 1998.

[PS03] F. Ponchio and M. Sala, *A lower bound on the distance of cyclic codes*, BCRI preprint, www.bcri.ucc.ie 7, University College Cork, Boole Centre BCRI, UCC Cork, Ireland, 2003.

[PS13] M. Piva and M. Sala, *A new bound for cyclic codes beating the roos bound*, Algebraic Informatics, Springer, 2013, pp. 101–112.

[PW72] W. W. Peterson and Jr. E. J. Weldon, *Error-correcting codes*, second ed., The M.I.T. Press, Cambridge, Mass.-London, 1972.

[Rei06] B. Reinert, *Gröbner bases in function rings—a guide for introducing reduction relations to algebraic structures*, Journal of Symbolic Computation **41** (2006), no. 11, 1264 – 1294.

[Roo82] C. Roos, *A generalization of the BCH bound for cyclic codes, including the Hartmann-Tzeng bound*, J. Combin. Theory Ser. A **33** (1982), no. 2, 229–232.

[Roo83] ______, *A new lower bound for the minimum distance of a cyclic code*, IEEE Trans. on Inf. Th. **29** (1983), no. 3, 330–332.

[Sal01] Massimiliano Sala, *On some algebraic methods for coding theory*, Ph.D. thesis, University of Milan, Milan, Italy, 2001.

[Sal02] M. Sala, *Gröbner bases and distance of cyclic codes*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 2, 137–162.

[Sal07] ______, *Gröbner basis techniques to compute weight distributions of shortened cyclic codes*, Journal of Algebra and Its Applications **6** (2007), no. 3, 403–404.

[Sch88] T. Schaub, *A linear complexity approach to cyclic codes*, Ph.D. thesis, Swiss Federal Inst. of Tech., Zurich, 1988.

[SWST96] K. K. Shen, C. Wang, B.-Z. Shen, and K. K. Tzeng, *Generation of matrices for determining minimum distance and decoding of cyclic codes*, IEEE Trans. on Inf. Th. **42** (1996), no. 2, 653–657.

[Tra88] C. Traverso, *Gröbner trace algorithms*, ISSAC, 1988, pp. 125–138.

[Var97a] A. Vardy, *Algorithmic complexity in coding theory and the minimum distance problem*, Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, 1997, pp. 92–109.

[Var97b] ______, *The intractability of computing the minimum distance of a code*, IEEE Trans. on Inf. Th. **43** (1997), no. 6, 1757–1766.

[vL95] J. H. van Lint, *Repeated-root cyclec codes*, IEEE Trans. on Inf. Th. **37** (1995), no. 2, 343–345.

[vLW86] J. H. van Lint and R. M. Wilson, *On the minimum distance of cyclic codes*, IEEE Trans. on Inf. Th. **32** (1986), no. 1, 23–40.

[Zac78] G. Zacharias, *Generalized Gröbner bases in commutative polynomial rings*, Ph.D. thesis, MIT, 1978.

[ZB12] A. Zeh and S. Bezzateev, *A new bound on the minimum distance of cyclic codes using small-minimum-distance cyclic codes*, Designs, Codes and Cryptography (2012), 1–18.

[ZK10] J. Zheng and T. Kaida, *An algorithm for new lower bound of minimum distance by dft for cyclic codes*, Information Theory and its Applications (ISITA), 2010 International Symposium on, IEEE, 2010, pp. 846–849.

[ZK11] ______, *On relationship between proposed lower bound and shift bound for cyclic codes*, Signal Design and its Applications in Communications (IWSDA), 2011 Fifth International Workshop on, IEEE, 2011, pp. 13–16.

[ZWZB12] Alexander Z., A. Wachter-Zeh, and S. V. Bezzateev, *Decoding cyclic codes up to a new bound on the minimum distance*, IEEE Trans. Inform. Theory **58** (2012), no. 6, 3951–3960.