A commutative algebra approach to linear codes

Marta Giorgetti (giorge@mat.unimi.it)
Department of Mathematics, University of Milano, Italy.

Massimiliano Sala (msala@bcri.ucc.ie)
Boole Centre for Research in Informatics, UCC Cork, Ireland

Abstract

Recently some methods have been proposed to find the distance and weight distribution of cyclic codes using Gröbner bases. We identify a class of codes for which these methods can be generalized. We show that this class contains all interesting linear codes (i.e., with $d \geq 2$) and we provide variants and improvements. This approach sometimes reveals an unexpected algebraic structure in the code. We also investigate the decoding for an interesting sub-class, proving the existence of general error locator polynomials.

Keywords: Linear code, distance, weight distribution, Gröbner basis, general error locator polynomial.

Researchers in Coding Theory have been extensively investigating error correcting codes with algebraic methods, since the very beginning of their theory (see for example the two classical books [PW72] and [MS77] and the recent survey [PHB98]). The algebraic approach has been successful in providing classes of codes that can be studied easily and that can be decoded with (relative) efficiency.

There are good reasons to single out one of these classes: the class of cyclic codes. First, cyclic codes enjoy a rich algebraic structure, permitting both fast sharp estimates on their most important parameters (see e.g. [BRC60],[HT74], [Roo83], [BS06],[ST00]) and the parameters' exact determination via commutative algebra techniques (see [Sal02], [MS03],[Sal06]).
Second, some subclasses of cyclic codes (such as the Reed-Solomon codes and the BCH codes) possess fast decoding algorithms ([Fit95]), so that most actual coding implementations (in hardware and software) are based on these codes. Third, in [OS05] a novel decoding algorithm has been introduced for generic cyclic codes, which has been shown experimentally to be extremely efficient (and a proof is given for some special cases in [MOS06] and [OS06]).

The aim of this paper is to view linear codes as a "generalization" of cyclic codes, trying to export techniques for a class to the whole set of codes. We

note that other researchers have tried similar generalizations. For example, in [PSvW91] it is shown that in some sense any linear code is an Algebraic-Geometry code and in [FL98] it is shown that any linear code can be seen as an affine-variety code.

The remainder of this paper is structured as follows:

- Section 1, we provide preliminaries and notation; we assume that the reader has some familiarity with standard algebra, such as finite field theory, but we assume he is not familiar with coding theory, so we give all necessary definitions. In particular, we recall the definition of general error locator polynomials.

- Section 2, we define our code family (the *nth-root codes*), provide some examples, give some results and show some applications. To be more precise, we prove that any interesting linear code (i.e., with $d \geq 2$) can be seen as an nth-root code, we describe some ideals whose varieties determine the distance and weight distribution (so that they can be computed via Gröbner basis techniques) and we give a similar algorithm to deal with cosets. We introduce several sub-families, including what we call *proper maximal zerofree* codes.

- Section 3, we extend the decoding techniques of [OS05] to a large subclass of nth-root codes, containing proper maximal zerofree codes. In this section the reader is assumed to have some understanding of Gröbner basis theory for 0-dimensional ideals. We prove that these codes admit general error locator polynomials of any type $\nu \geq 0$. We do so by describing precisely the geometric conditions behind the main results of [OS05] and introducing a special class of zero-dimensional ideals, which we call *stratified ideals*. From the shape of the Groebner basis of any stratified ideal, the existence of general error locator polynomials easily follows. Furthermore, we propose an alternative approach for the computation of said polynomials, which works better in some cases.

- Section 4, we see how well-known code families can be described as nth-root codes. This section is divided into sub-sections, any requiring some specific knowledge of the matter hereby exposed: cyclic codes, classical Goppa codes, RM codes, Goppa AG codes (with a detailed analysis for codes from Hermitian curves).

- Section 5, we provide some complexity considerations and we show some methods to accelerate the involved calculations. We apply the theory of semi-regular sequences to formally determine the regularity degree (aymptotically) for an ideal that is proved equivalent to our previously constructed, in the binary zero-free case. This requires the combinatorial valuation of some spurious solutions.

- Section 6, we summarize our results and point out to future research.

1 Preliminaries

In this section we fix some notation and we recall some basic concepts about general algebra, polynomial rings and linear codes.

We denote by $\mathbb{F}_q$ the finite field with q elements, where q is a power of a prime and by n a natural number such that q and n are relatively prime, $(n, q) = 1$. Let $1 \leq k \leq N \leq n$, $k, N \in \mathbb{N}$. We refer to the vector space of dimension N over $\mathbb{F}_q$ as to $(\mathbb{F}_q)^N$.

We use the symbol $\bigsqcup_{i\in I} B_i$ to denote the disjoint union of sets B_i, $i \in I$.

The zeros of polynomial $x^n - 1$, which are called *n-th roots of unity*, lie in an extension field $\mathbb{F}_{q^m}$ and in no smaller field. We denote the set of all these roots by R_n and they form a cyclic subgroup of $\mathbb{F}_{q^m}$, i.e. there is $\alpha \in \mathbb{F}_{q^m}$, called a *primitive* n-th root of unity, such that:

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

If $n = q^m - 1$, the zeros of $x^n - 1$ form the multiplicative group of field $\mathbb{F}_{q^m}$.

From now on q,n, k, N, m and α are understood (unless otherwise stated).

1.1 Polynomial rings

Let $\mathbb{K}$ be a field, let $\overline{\mathbb{K}}$ be the algebraic closure of $\mathbb{K}$ and let J be an ideal in polynomial ring $\mathbb{K}[Y] = \mathbb{K}[y_1, \ldots, y_s]$, with $s \geq 1$.

Definition 1.1. *Given a polynomial $f \in \mathbb{K}[Y]$, we denote by $\mathcal{V}(f)$ the set of all zeros of f in $(\overline{\mathbb{K}})^s$, i.e.*

$$\mathcal{V}(f) = \{(a_1, \ldots, a_s) \in (\overline{\mathbb{K}})^s \mid f(a_1, \ldots, a_s) = 0\}.$$

Given an ideal $J \subseteq \mathbb{K}[Y]$, we denote by $\mathcal{V}(J)$ the set of zeros of J in $(\overline{\mathbb{K}})^s$, i.e.

$$\mathcal{V}(J) = \{(a_1, \ldots, a_s) \in (\overline{\mathbb{K}})^s \mid f(a_1, \ldots, a_s) = 0, \forall f \in J\}.$$

Definition 1.2. *Let $S \subseteq (\overline{\mathbb{K}})^s$. Then the set of all polynomials $f \in \mathbb{K}[Y]$ such that $f(a_1, \ldots, a_s) = 0$ for all points $(a_1, \ldots, a_s) \in S$ forms an ideal in $\mathbb{K}[Y]$. This ideal is the* **vanishing** *ideal of S and is denoted by $\mathcal{I}(S)$.*

Let $L \subset \mathbb{K}[Y]$, we denote by $< L >$ the ideal in $\mathbb{K}[Y]$ generated by L.

1.2 Coding Theory

Definition 1.3. *Let H be an $(N-k)\times N$ matrix with entries in $\mathbb{F}_{q^m}$, such that its rank over $\mathbb{F}_q$ is $N-k$. The set C of all vectors $c \in (\mathbb{F}_q)^N$ such that $Hc^T = 0$ is an (N, k)* **linear** *code over $\mathbb{F}_q$, N is the* **length** *and k is the* **dimension**.

The elements of C are called **codewords** *and matrix H is a* **parity-check** *matrix of C. If $q = 2$, C is called a binary code. Any $k \times N$ matrix G whose rows form a vector basis of C is called a* **generator matrix** *of C.*

Definition 1.4. *Let x, y be two vectors in $(\mathbb{F}_q)^N$. Then:*

(1) the Hamming **distance** *$d(x, y)$ between x and y is the number of coordinates in which x and y differ;*

(2) the Hamming **weight** *$\mathrm{w}(x)$ is the number of nonzero components of x.*

Definition 1.5. *Let C be a linear code. The number*

$$d_C = \min_{x,y \in C, x \neq y} d(x, y) = \min_{x \in C, x \neq 0} \mathrm{w}(c)$$

is called the (minimum) **distance** *of C.*

From now on, "code" means "linear code".

If a code C has length N, dimension k and distance d, we say that C is an $[N, k, d]$ code.

When a codeword is transmitted, it can be affected by errors or erasures. An error occurs when one codeword component is changed into another field element and an erasure occurs when the received component has an unknown value. We know where the erasures are (erasure locations), but we do not know where the errors occur (error locations). It is convenient to collect the errors in a vector which is the received vector minus the sent word.
If there is a decoding procedure for C that can always correct τ errors or less, then we say that the error correction capability of the code C is τ. We denote by t the maximum value for τ. It is known that $t = \lfloor \frac{d-1}{2} \rfloor$.
Moreover, for any ν and τ natural numbers such that $2\tau + \nu < d$, we know that C can correct simultaneously ν erasures and τ errors.

Definition 1.6. *Let C be an (N, k) code. We denote by $A_i = A_i(C)$ the number of words in C with weight i. The integer set $\{A_0, A_1, \ldots, A_N\}$ is called the weight* **distribution** *of C.*

Definition 1.7. *Let $C \subseteq (\mathbb{F}_q)^N$ be an (N, k) code. For any vector $a \in (\mathbb{F}_q)^N$ the set*

$$a + C = \{a + x : x \in C\}$$

in called a **coset** *(or translate) of C. Let H be a parity-check matrix of C. Then vector $S(y) = Hy^T$ of length $N - k$ is called the* **syndrome** *of y. We denote by $A_i(a + C)$ the number of vectors of weight i in translate $a + C$.*

Definition 1.8. *Let C be an (N, k) code over $\mathbb{F}_q$ with parity-check matrix H. Let D be a proper subset of $\mathcal{N} = \{1, \ldots, N\}$. Let H' be the matrix obtained from H by deleting columns $h_{.,j}$, $j \in D$. We define the* **shortened** *code $C(D)$ as the code having H' as a parity-check matrix.*

1.3 General error locator polynomial

Let C be an $[N, k, d]$ code over $\mathbb{F}_q$, t its correction capability and H a parity-check matrix. Let $d \geq 3$. The syndromes lie in $(\mathbb{F}_{q^m})^{N-k}$ and form a vector space of dimension $(N-k)$ over $\mathbb{F}_q$. Let α be a primitive N-th root of unity in $\mathbb{F}_{q^m}$, so that $n = N$. Let $r = N - k$.

Definition 1.9. *Let $\mathcal{L}_C$ be a polynomial in $\mathbb{F}_q[X, z]$, where $X = (x_1, \ldots, x_r)$. Then $\mathcal{L}_C$ is a* **general error locator polynomial** *of C if*

(1) $\mathcal{L}_C(X, z) = z^t + a_{t-1}z^{t-1} + \cdots + a_0$, with $a_j \in \mathbb{F}_q[X]$, $0 \leq j \leq t-1$, that is, $\mathcal{L}_C$ is a monic polynomial with degree t with respect to the variable z and its coefficients are in $\mathbb{F}_q[X]$;

(2) given a syndrome $\mathbf{s} = (\overline{s}_1, \ldots \overline{s}_r) \in (\mathbb{F}_{q^m})^{N-k}$, corresponding to a vector error of weight $\mu \leq t$ and error locations $\{k_1, \ldots, k_\mu\}$, if we evaluate the X variables in $\mathbf{s}$, then the roots of $\mathcal{L}_C(\mathbf{s}, z)$ are exactly $\{\alpha^{k_1}, \ldots, \alpha^{k_\mu}, \underbrace{0, \ldots, 0}_{t-\mu}\}$.

Given a generic code C, the existence of a general error locator polynomial is not known. In [OS05] the authors prove its existence for any cyclic code.

We can extend Definition 1.9 to the case when there are also erasures.

Definition 1.10. *Let $\mathcal{L}$ be a polynomial in $\mathbb{F}_q[X, W, z]$, $X = (x_1, \ldots, x_r)$ and $W = (w_\nu, \ldots, w_1)$, where ν is the number of erasures that occurred. Then $\mathcal{L}$ is a* **general error locator polynomial of type** *ν of C if*

(1) $\mathcal{L}(X, W, z) = z^\tau + a_{\tau-1}z^{\tau-1} + \cdots + a_0$, with $a_j \in \mathbb{F}_q[X, W]$, for any $0 \leq j \leq \tau - 1$, that is, $\mathcal{L}$ is a monic polynomial with degree τ in the variable z and coefficients in $\mathbb{F}_q[X, W]$;

(2) for any syndrome $\mathbf{s} = (\overline{s}_1, \ldots, \overline{s}_r)$ and any erasure location vector $\mathbf{w} = (\overline{w}_1, \ldots, \overline{w}_\nu)$, corresponding to an error of weight $\mu \leq \tau$ and error locations $\{k_1, \ldots, k_\mu\}$, if we evaluate the X variables in $\mathbf{s}$ and the W variables in $\mathbf{w}$, then the roots of $\mathcal{L}(\mathbf{s}, \mathbf{w}, z)$ are $\{\alpha^{k_1}, \ldots, \alpha^{k_\mu}, \underbrace{0, \ldots, 0}_{\tau-\mu}\}$.

If such $\mathcal{L}$ exists for a given code C, then we name the polynomial $\mathcal{L}_C^\nu$.

To be consistent with our notation, we refer to $\mathcal{L}_C$ also as to a **general locator polynomial of type 0**, where clearly $\mathcal{L}_C = \mathcal{L}_C^0$.

For a code C, the possession of a general locator polynomial $\mathcal{L}_C^\nu$ of type ν for all $0 \leq \nu < d$ might be a stronger condition than the possession of a general error locator polynomial $\mathcal{L}_C$, but in [OS05] the authors prove that any cyclic code admits a general locator polynomial of type ν, for $0 \leq \nu < d$.

2 General nth-root codes

2.1 Definition and first properties

Definition 2.1. *Let $L \subset R_n \cup \{0\}$, $L = \{l_1, \ldots, l_N\}$ and $\mathcal{P} = \{g_1(x), \ldots, g_r(x)\}$ in $\mathbb{F}_{q^m}[x]$ such that $\forall i = 1, \ldots, N$ there is at least one $j = 1, \ldots, r$ such that $g_j(l_i) \neq 0$. We denote by $C = \Omega(q, n, q^m, L, \mathcal{P})$ the code defined over $\mathbb{F}_q$ having*

$$H = \begin{pmatrix} g_1(l_1) & \ldots & g_1(l_N) \\ g_2(l_1) & \ldots & g_2(l_N) \\ \vdots & & \vdots \\ g_r(l_1) & \ldots & g_r(l_N) \end{pmatrix} = \begin{pmatrix} g_1(L) \\ g_2(L) \\ \vdots \\ g_r(L) \end{pmatrix}$$

*as its parity-check matrix. We say that C is an **nth-root code**.*

Remark 2.2. Code $C = \Omega(q, n, q^m, L, \mathcal{P})$ is linear over $\mathbb{F}_q$, its length is $N = |L|$ and its distance d is greater than or equal to 2, because there are no columns in H composed only of zeros.

If $0 \in L$ we assume $l_N = 0$ (any re-ordering of L gives an equivalent code). We will denote by $\bar{L}$ the set $R_n \setminus L$.

Definition 2.3. *Let $C = \Omega(q, n, q^m, L, \mathcal{P})$ be an nth-root code and $v \in (\mathbb{F}_q)^N$.*
*If $\bar{L} = \emptyset$, we say that C is **maximal**.*
*If $\mathcal{P} \subset \mathbb{F}_q[x]$, we say that C is **proper**.*
*If $0 \notin L$, we say that C is **zerofree**, non-zerofree otherwise.*
*Vector v is **zerofree** if either C is zerofree or C is non-zerofree but $v_N = 0$.*

Since any function from $\mathbb{F}_{q^m}$ to itself can be expressed as a polynomial, we can accept in $\mathcal{P}$ also rational functions of type f/g, $f, g \in \mathbb{F}_{q^m}$, such that $g(\bar{x}) \neq 0$ for any $\bar{x} \in \mathbb{F}_{q^m}$. We do so from now on, without further comments.

Example 2.4. Let $q = 2$, $n = 7$, $q^m = 8$, $L = \mathbb{F}_{2^3} =< \beta > \cup \{0\}$ and $\mathcal{P} = \{g_1(x) = \frac{1}{x^2+x+1}, g_2(x) = \frac{x}{x^2+x+1}\}$. The seven 7th roots of unity are all the elements of $\mathbb{F}_8^*$, $R_7 = \mathbb{F}_8^*$. The nth-root code $C = \Omega(2, 7, 8, \mathbb{F}_8, \{g_1, g_2\})$ is non-zerofree ($0 \in L$), maximal ($\bar{L} = R_n \setminus L = \emptyset$), proper (both g_1 and g_2 lie in $\mathbb{F}_2[x]$) and its parity-check matrix is the following:

$$H = \begin{pmatrix} g_1(1) & g_1(\beta) & g_1(\beta^2) & g_1(\beta^3) & g_1(\beta^4) & g_1(\beta^5) & g_1(\beta^6) & g_1(0) \\ g_2(1) & g_2(\beta) & g_2(\beta^2) & g_2(\beta^3) & g_2(\beta^4) & g_2(\beta^5) & g_2(\beta^6) & g_2(0) \end{pmatrix}, \text{ i.e.}$$

$$H = \begin{pmatrix} 1 & \beta^2 & \beta^4 & \beta^2 & \beta & \beta & \beta^4 & 1 \\ 1 & \beta^3 & \beta^6 & \beta^5 & \beta^5 & \beta^6 & \beta^3 & 0 \end{pmatrix}.$$

It is easy to see that C is an [8,2,5] code with generator matrix

$$G = \begin{pmatrix} 1\,0\,0\,1\,0\,1\,1\,1 \\ 0\,1\,1\,1\,1\,1\,1\,0 \end{pmatrix},$$

and weight distribution $\{A_0 = 1, A_1 = A_2 = A_3 = A_4 = 0, A_5 = 2, A_6 = 1\}$.

In the next example we show that not all maximal nth-root codes can be seen as maximal proper nth-root codes.

Example 2.5. Let $q = 2$, $n = 5$, $q^m = 2^4$, $L = R_5$ and $\mathcal{P} = \{g\}$, where $g = \gamma^{12}x^4 + \gamma^{11}x^3 + x^2 + \gamma^{14}x + \gamma^3$ and γ is a primitive element of $\mathbb{F}_{16}$. Let $C = \Omega(2, 5, 2^4, R_5, \mathcal{P})$. Code C is maximal ($\bar{L} = \emptyset$) and zero-free ($0 \notin L$) and its parity-check matrix is the following:

$$H = \left(g(\gamma^3), g(\gamma^6), g(\gamma^9), g(\gamma^{12}), g(\gamma^{15})\right) = \left(\gamma^6, \gamma^2, \gamma^3, \gamma^{14}, \gamma^{15}\right).$$

It is easy to see that C is an [5,2,3] code with generator matrix

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

We show that C is not proper maximal by contradiction. If C is (zerofree) proper maximal then $C = \Omega(2, 5, 2^4, R_5, \mathcal{P}')$, where $\mathcal{P}' = \{g'_1, \ldots, g'_r\} \subset \mathbb{F}_2[x]$ for some $r \geq 1$. Its parity-check matrix is then

$$H' = \begin{pmatrix} g_1(\gamma^3), & g_1(\gamma^6), & g_1(\gamma^9), & g_1(\gamma^{12}), & g_1(\gamma^{15}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_r(\gamma^3), & g_r(\gamma^6), & g_r(\gamma^9), & g_r(\gamma^{12}), & g_r(\gamma^{15}) \end{pmatrix}.$$

Let $e_1 = g'(\gamma^3)$, $e_2 = g'(\gamma^6)$, $e_3 = g'(\gamma^9)$, $e_4 = g'(\gamma^{12})$, $e_5 = g'(\gamma^{15})$, where $g'(x) = g_i(x)$ for some $i = 1, \ldots, r$. Since $(e_1, e_2, e_3, e_4, e_5)$ is a row of H, it must satisfy $e_1 + e_2 + e_3 = 0$ and $e_3 + e_4 + e_5 = 0$. In other words, the following ideal $J \subset \mathbb{F}_{16}[b_0, \ldots, b_{15}, e_1, \ldots, e_5]$ has at least a solution $\varepsilon = (\bar{b}_0, \ldots, \bar{b}_{15}, \bar{e}_1, \ldots, \bar{e}_5)$ in $\mathcal{V}(J)$ such that $(\bar{e}_1, \bar{e}_2, \bar{e}_3, \bar{e}_4, \bar{e}_5) \neq (0, 0, 0, 0, 0)$,

$$\begin{aligned} J =< & e_1 + e_2 + e_3,\ e_3 + e_4 + e_5,\ \{b_i^2 + b_i\}_{0 \leq i \leq 15},\ \{e_i^{16} + e_i\}_{1 \leq i \leq 5}, \\ & g'(\gamma^3) - e_1,\ g'(\gamma^6) - e_2,\ g'(\gamma^9) - e_3,\ g'(\gamma^{12}) - e_4,\ g'(\gamma^{15}) - e_5 >, \end{aligned}$$

where $g' = \sum_0^{15} b_i x^i \in \mathbb{F}_2[x]$. A computer computation shows that a Gröbner basis of J contains $\{e_1, \ldots, e_5\}$ and so $\mathcal{V}(J)$ does not contain ε, hence g' does not exist. This means that no polynomial in $\mathcal{P}$ can have coefficients in $\mathbb{F}_2$, which proves our claim.

Remark 2.6. In order to define the same nth-root code it is possible to use different n. For example to define a linear code with length $N = 5$, we can use the five 5th roots of unity or five elements chosen from the set of the seven 7th roots of unity. See next example.

Example 2.7. Let C be a linear code over $\mathbb{F}_2$ having parity-check matrix

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

It is possible to view C as a maximal, zerofree nth-root code $\Omega(2, 5, 2^4, L_1, \mathcal{P}_1)$, where $L_1 = R_5 = \{\gamma^3, \gamma^6, \gamma^9, \gamma^{12}, \gamma^{15}\} \subset \mathbb{F}_{16} =< \gamma > \cup\{0\}$ and $\mathcal{P}_1 \subset \mathbb{F}_{16}[x]$ is $\mathcal{P}_1 = \{g_1, g_2\}$, with

$$g_1 = \gamma^7 x^4 + \gamma^{14} x^3 + \gamma^{11} x^2 + \gamma^{13} x + 1, \qquad g_2 = \gamma^2 x^4 + \gamma^4 x^3 + \gamma x^2 + \gamma^8 x + 1\ .$$

Code C can also be seen as a non-maximal nth-root code zerofree with different parameters, that is, $C = \Omega(2, 7, 2^3, L_2, \mathcal{P}_2)$, where $L_2 \subset R_7 = \mathbb{F}_8^* =< \beta >$, $L_2 = \{\beta, \beta^2, \beta^3, \beta^4, \beta^5\}$ and $\mathcal{P}_2 \subset \mathbb{F}_{2^3}[t]$ is $\mathcal{P}_2 = \{p_1, p_2\}$, with

$$p_1 = t^4 + t^2 + t + 1, \qquad p_2 = \beta^4 t^4 + \beta^6 t^3 + t + \beta^2\ .$$

Moreover, code C can also be seen as a non-maximal, non-zerofree nth-root code with the following parameters: $C = \Omega(2, 7, 2^3, L_3, \mathcal{P}_3)$, with $L_3 \subset \mathbb{F}_8$, $L_3 = \{\beta, \beta^2, \beta^3, \beta^4, 0\}$ and $\mathcal{P}_3 \subset \mathbb{F}_8[z]$ is $\mathcal{P}_3 = \{h_1, h_2\}$, where

$$h_1 = \beta^5 z^4 + z^3 + \beta^5 z^2 + \beta^4 z, \qquad h_2 = \beta^6 z^4 + \beta^3 z^2 + \beta^5 z + 1\ .$$

Note however that code C cannot be seen as a maximal non-zerofree code.

The next proposition shows in particular that any correctable code can be seen as an nth-root code for suitable values of n.

Proposition 2.8. *Let C be a code over $\mathbb{F}_q$ of length N and $d \geq 2$. Then C is an nth-root code for any $n \geq N - 1$ such that $(n, q) = 1$. In particular:*

(1) if $n = N$, then C can be maximal zerofree,

(2) if $n = N - 1$, then C is maximal non-zerofree.

Proof. Let C be a linear code over $\mathbb{F}_q$ of length N, dimension k and $d \geq 2$, with parity-check matrix $H = (h_{i,j}) \in (\mathbb{F}_q)^{(N-k)\times N}$. Since $d \geq 2$ there is no $j = 1, \ldots, N$ such that $h_{i,j} = 0$, $\forall i = 1, \ldots N - k$. Let n be a natural number such that $n \geq N - 1$ and $(n, q) = 1$. Let $R_n = \{\alpha_1, \ldots, \alpha_n\}$ be the set of nth-roots of unity over $\mathbb{F}_q$.

- Suppose that $n \geq N$. Let L be a subset of R_n, $|L| = N$, and $r = N - k$. Thanks to the Lagrange interpolation theorem we can find r polynomials $g_i(x) \in \mathbb{F}_{q^m}[x]$ such that $g_i(\alpha_j) = h_{i,j}\, \forall\, \alpha_j \in L$, $i = 1, \ldots, r$, $j = 1, \ldots, N$,

viewing any $h_{i,j}$ as an element of $\mathbb{F}_{q^m}$. We collect polynomials $g_i(x)$ in set $\mathcal{P} = \{g_i\}_{1\le i\le r}$. Polynomials $g_i(x)$ are such that for any $i = 1, \cdots, r$ there is at least one $1 \le j \le r$ such that $g_j(\alpha_i) \neq 0$. Then it is obvious that code C can be seen as the zerofree nth-root code $\Omega(q, n, q^m, L, \mathcal{P})$.

- With the above construction, if $n = N$ code C is maximal, since $L = R_n$.
- Let L be a set composed of 0 and $N-1$ elements of R_n. With the above argument it is easy to proof that C is a non-zerofree nth-root code.
  If $n = N-1$, code C is maximal non-zerofree, since $L = R_n \cup \{0\}$.

□

Corollary 2.9. *Let C be a code. C is an nth-root code if and only if $d \ge 2$.*

Proof. It follows immediately from Proposition 2.8 and from Remark 2.2. □

Thanks to previous proposition, an (linear) $[N, k, d]$ code C, $d \ge 2$, can be seen as an nth-root code, but we do not know whether it can be seen as a proper nth-root code: we only know that there are codes that cannot be seen as *maximal* proper nth-root (see Example 2.5).

2.2 Computing distance and weight distribution for an nth-root code

In this section we provide a method to compute the distance and weight distribution of a code C, given a representation of C as an nth-root code.

The following two ideals are necessary to our purposes.

Definition 2.10. *Let $C = \Omega(q, n, q^m, L, \mathcal{P})$ be an nth-root code, w and $\hat{w}$ be natural numbers such that $2 \le w \le N = |L|$, $1 \le \hat{w} \le N-1$. We denote by $J_w(C)$ and $\hat{J}_{\hat{w}}(C)$ the following two ideals:*

$$J_w = J_w(C) = J_w(q, n, q^m, L, \mathcal{P}) \subset \mathbb{F}_{q^m}[z_1, \ldots, z_w, y_1, \ldots, y_w],$$
$$\hat{J}_{\hat{w}} = \hat{J}_{\hat{w}}(C) = \hat{J}_{\hat{w}}(q, n, q^m, L, \mathcal{P}) \subset \mathbb{F}_{q^m}[z_1, \ldots, z_{\hat{w}}, y_1, \ldots, y_{\hat{w}}, \nu],$$

$$\begin{aligned} J_w = \langle \quad & \left\{\textstyle\sum_{h=1}^{w} y_h g_s(z_h)\right\}_{1\le s\le r}, \left\{y_j^{q-1}-1\right\}_{1\le j\le w}, \\ & \left\{p_{ij}(z_i, z_j)\right\}_{1\le i<j\le w}, \left\{\tfrac{z_j^n-1}{\prod_{l\in\bar{L}}(z_j-l)}\right\}_{1\le j\le w} \quad \rangle \end{aligned} \tag{1}$$

$$\begin{aligned} \hat{J}_{\hat{w}} = \langle \quad & \left\{\textstyle\sum_{h=1}^{\hat{w}} y_h g_s(z_h) + \nu g_s(0)\right\}_{1\le s\le r}, \left\{y_j^{q-1}-1\right\}_{1\le j\le \hat{w}} \\ & \nu^{q-1}-1, \left\{p_{ij}(z_i, z_j)\right\}_{1\le i<j\le \hat{w}}, \left\{\tfrac{z_j^n-1}{\prod_{l\in\bar{L}}(z_j-l)}\right\}_{1\le j\le \hat{w}} \quad \rangle \end{aligned} \tag{2}$$

where $p_{ij} = \sum_{h=0}^{n-1} z_i^h z_j^{n-1-h} = \frac{z_i^n - z_j^n}{z_i - z_j}$ are in $\mathbb{F}_q[z_i, z_j]$.
We denote by $\eta(J_w)$ and $\hat{\eta}(\hat{J}_{\hat{w}})$ the integers $\eta(J_w) = |\mathcal{V}(J_w)|$, $\hat{\eta}(\hat{J}_{\hat{w}}) = |\mathcal{V}(\hat{J}_{\hat{w}})|$.

Remark 2.11. Ideals J_w and $\hat{J}_{\hat{w}}$ are radical, since they contain polynomials $y_j^q - y_j$ and $z_j^{n+1} - z_j$ with $j = 1, \ldots, w$ for J_w and $j = 1, \ldots, \hat{w}$ for $\hat{J}_{\hat{w}}$ ([Sei74]).
Remark 2.12. If we are in the binary case ($q = 2$), variables y_j, $j = 1, \ldots, w$, and ν are 1, and so we can omit them and the ideals become:

$$J_w = J_w(C) = J_w(2, n, 2^m, L, \mathcal{P}) \subset \mathbb{F}_{2^m}[z_1, \ldots, z_w],$$

$$J_w = \langle \{\textstyle\sum_{h=1}^{w} g_s(z_h)\}_{1 \le s \le r}, \{p_{ij}(z_i, z_j)\}_{1 \le i < j \le w} \left\{ \frac{z_j^n - 1}{\prod_{l \in \bar{L}}(z_j - l)} \right\}_{1 \le j \le w} \rangle;$$

$$\hat{J}_{\hat{w}} = \hat{J}_{\hat{w}}(C) = \hat{J}_{\hat{w}}(2, n, 2^m, L, \mathcal{P}) \subset \mathbb{F}_{2^m}[z_1, \ldots, z_{\hat{w}}],$$

$$\hat{J}_{\hat{w}} = \langle \left\{ \textstyle\sum_{h=1}^{\hat{w}} g_s(z_h) + g_s(0) \right\}_{1 \le s \le r}, \{p_{ij}(z_i, z_j)\}_{1 \le i < j \le \hat{w}}, \left\{ \frac{z_j^n - 1}{\prod_{l \in \bar{L}}(z_j - l)} \right\}_{1 \le j \le \hat{w}} \rangle.$$

Proposition 2.13. *Let $C = \Omega(q, n, q^m, L, \mathcal{P})$ be an nth-root code. In the zerofree case, there is at least one codeword of weight w in C if and only if there exists at least one solution of $J_w(C)$. In the non-zerofree case, there is at least one codeword of weight w in C if and only if there exists at least one solution of $J_w(C)$ or of $\hat{J}_{w-1}(C)$. Moreover the number of codewords of weight w is*

$$A_w = \tfrac{\eta(J_w)}{w!} \quad \textit{in the zerofree case and}$$
$$A_w = \tfrac{\eta(J_w)}{w!} + \tfrac{\hat{\eta}(\hat{J}_{w-1})}{(w-1)!} \quad \textit{in the non-zerofree case}$$

The proof is reported in Subsection 2.5.

2.3 Algorithms

Since the number of solutions of an ideal J is directly computed from any Gröbner basis of J (see [BCRT93]), we can easily describe an algorithm to compute the weight distribution (and the distance) of an nth-root code, by applying Proposition 2.13.

We first consider the zerofree case.

INPUT:	a zerofree nth-root code $C = \Omega(q, n, q^m, L, \mathcal{P})$,
	an integer $2 \le w \le \lvert L \rvert$
OUTPUT:	the element A_w of the weight distribution of C
STEP 1:	construct ideal $J_w = J_w(C)$
STEP 2:	compute a Gröbner basis $\mathcal{G}_w$ of J_w
STEP 3:	use $\mathcal{G}_w$ to get the number $\eta(J_w)$ of points in $\mathcal{V}(J_w)$
STEP 4:	return $\frac{\eta(J_w)}{w!}$

We now consider the non-zerofree case.

INPUT:	a non-zerofree nth-root code $C = \Omega(q, n, q^m, L, \mathcal{P})$,
	an integer $2 \leq w \leq \lvert L \rvert$
OUTPUT:	the element A_w of the weight distribution of C
STEP 1:	construct ideals $J_w = J_w(C)$ and $\hat{J}_{w-1} = \hat{J}_{w-1}(C)$
STEP 2:	compute a Gröbner basis $\mathcal{G}_w$ of J_w and
	compute a Gröbner basis $\hat{G}_{w-1}$ of $\hat{J}_{w-1}$
STEP 3:	use $\mathcal{G}_w$ to get the number $\eta(J_w)$ of points in $\mathcal{V}(J_w)$ and
	use $\hat{G}_{w-1}$ to get the number $\hat{\eta}(\hat{J}_{w-1})$ of points in $\mathcal{V}(\hat{J}_{w-1})$
STEP 4:	return $\frac{\eta(J_w)}{w!} + \frac{\hat{\eta}(\hat{J}_{w-1})}{(w-1)!}$

We give an example for the non-zerofree case.

Example 2.14. Consider the nth-root code C as in Example 2.4. We compute its weight distribution by using our algorithm. Setting $w = 2$ we construct ideals $J_2(C) \subseteq \mathbb{F}_2[z_1, z_2]$ and $\hat{J}_1(C) \subseteq \mathbb{F}_2[z_1]$:

$$J_2(C) =< g_1(z_1) + g_1(z_2), g_2(z_1) + g_2(z_2), z_1^7 - 1, z_2^7 - 1, p(z_1, z_2) >$$

$$\hat{J}_1(C) =< g_1(z_1) + g_1(0), g_2(z_1) + g_2(0), z_1^7 - 1 >$$

Their Gröbner bases $\mathcal{G}_2$ and $\hat{\mathcal{G}}_1$ are trivial and hence there are no words of weight 2 in this nth-root code. The same happens for $w = 3$ and $w = 4$, so that $A_3 = A_4 = 0$. Setting $w = 5$ we construct the ideals J_5 and $\hat{J}_4$. Basis $\mathcal{G}_5$ is trivial, but basis $\hat{\mathcal{G}}_4$ has the following leading terms

$$\left\{ z_1 z_2,\ z_1^2,\ z_1 z_3^2,\ z_2^3,\ z_1 z_4^3,\ z_3^4,\ z_2^2 z_3^2,\ z_4^5,\ z_2^2 z_4^3,\ z_3^3 z_4^3 \right\}.$$

These monomials permit us to compute the number $\hat{\eta}(\hat{J}_4) = 48$ ([BCRT93]). We get $A_5 = \frac{\eta(J_5)}{5!} + \frac{\hat{\eta}(\hat{J}_4)}{4!} = \frac{48}{4!} = 2$ (note that the 2 words of weight 5 in C have the last component non zero). Computing $\mathcal{G}_6$ we have a non trivial result, $\eta(J_6) = 720$, and for $\hat{J}_5$ we get an empty variety. The words of weight 6 are then $A_6 = \frac{\eta(J_6)}{6!} + \frac{\hat{\eta}(\hat{J}_5)}{5!} = \frac{720}{6!} = 1$. Summarizing, we have:

w	$\mathcal{G}(J_w)$	$\hat{\mathcal{G}}(\hat{J}_{w-1})$	$\eta(J_w)$	$\hat{\eta}(\hat{J}_{w-1})$	A_w
2,3,4,7	$\{1\}$	$\{1\}$	0	0	0
5	$\{1\}$	not trivial	0	48	2
6	not trivial	$\{1\}$	720	0	1
8	–	$\{1\}$	–	0	0

2.4 Computing the weight distribution for cosets

In this subsection we study the computation of the weight distribution of translates of any nth-root code $C = \Omega(q, n, q^m, L, \mathcal{P})$.

We can define two ideals and use them to calculate the weight distribution of cosets, similarly to what is done in previous sections.

Definition 2.15. *Let $C = \Omega(q, n, q^m, L, \mathcal{P})$ be an nth-root code, w and $\hat{w}$ be natural numbers such that $2 \leq w \leq N = |L|$, $1 \leq \hat{w} \leq N-1$. Let $a \in (\mathbb{F}_q)^N \setminus C$ and $\sigma(a) \in (\mathbb{F}_{q^m})^r$ be its syndrome. We denote by $J_w(a+C)$ and $\hat{J}_{\hat{w}}(a+C)$ the following two ideals:*
$J_w(a+C) \subset \mathbb{F}_{q^m}[z_1, \ldots, z_w, y_1, \ldots, y_w]$, $\hat{J}_{\hat{w}}(a+C) \subset \mathbb{F}_{q^m}[z_1, \ldots, z_{\hat{w}}, y_1, \ldots, y_{\hat{w}}, \nu]$,

$$J_w(a+C) = \langle \begin{array}{l} \{\sum_{h=1}^{w} y_h g_s(z_h) - \sigma(a)\}_{1\leq s\leq r}, \{y_j^{q-1}-1\}_{1\leq j\leq w}, \\ \{p_{ij}(z_i, z_j)\}_{1\leq i<j\leq w}, \left\{\frac{z_j^n-1}{\prod_{l\in\bar{L}}(z_j-l)}\right\}_{1\leq j\leq w} \rangle; \end{array} \tag{3}$$

$$\hat{J}_{\hat{w}}(a+C) = \langle \begin{array}{l} \left\{\sum_{h=1}^{\hat{w}} y_h g_s(z_h) + \nu g_s(0) - \sigma(a)\right\}_{1\leq s\leq r}, \{y_j^{q-1}-1\}_{1\leq j\leq \hat{w}} \\ \nu^{q-1}-1, \{p_{ij}(z_i, z_j)\}_{1\leq i<j\leq \hat{w}}, \left\{\frac{z_j^n-1}{\prod_{l\in\bar{L}}(z_j-l)}\right\}_{1\leq j\leq \hat{w}} \rangle. \end{array} \tag{4}$$

We also define $\eta(J_w(a+C)) = |\mathcal{V}(J_w(a+C))|$, $\hat{\eta}(\hat{J}_{\hat{w}}(a+C)) = |\mathcal{V}(\hat{J}_{\hat{w}}(a+C))|$.

We have the following proposition.

Proposition 2.16. *Let $C = \Omega(q, n, q^m, L, \mathcal{P})$, $a \in (\mathbb{F}_q)^N \setminus C$, and $a + C$ a coset of code C. In the zerofree case, there is at least one vector of weight w in coset $a+C$ if and only if there is at least one solution of $J_w(a+C)$. In the non-zerofree case, there is at least one vector of weight w in $a+C$ if and only if there is at least one solution of $J_w(a+C)$ or of $\hat{J}_{w-1}(a+C)$. Furthermore, the number of vectors of weight w in $a+C$ is*

$$A_w(a+C) = \frac{\eta(J_w(a+C))}{w!} \quad \text{in the zerofree case and}$$
$$A_w(a+C) = \frac{\eta(J_w(a+C))}{w!} + \frac{\hat{\eta}(\hat{J}_{w-1}(a+C))}{(w-1)!} \quad \text{in the non-zerofree case}$$

Proof. We apply similar arguments to those of Proposition 2.13. □

Example 2.17. We consider code C as in Example 2.4. We know that it has 64 cosets (including C itself), any having a syndrome vectors $\sigma(a)$ in $(\mathbb{F}_8)^6$. Since H has two rows, we can consider syndromes in $(\mathbb{F}_8)^2$.
Let $a = (0,0,0,0,0,0,0,1)$ and $Ha^T = \sigma(a) = (1,0)^T$.
We construct ideals $J_w(a+C)$, $\hat{J}_{w-1}(a+C)$ $(2 \leq w \leq 8)$, we compute their Gröbner bases $\mathcal{G}_w^a = \mathcal{G}(J_w(a+C))$, $\hat{\mathcal{G}}_{w-1}^a = \hat{\mathcal{G}}(\hat{J}_{w-1}(a+C))$, obtaining the following results:

w	$\mathcal{G}_w^a$	$\hat{\mathcal{G}}_{w-1}^a$	$\eta(J_w(a+C))$	$\hat{\eta}(\hat{J}_{w-1}(a+C))$	$A_w(a)$
2,3,5,6	$\{1\}$	$\{1\}$	0	0	0
4	non trivial	$\{1\}$	48	0	2
7	$\{1\}$	non trivial	0	720	1
8	–	$\{1\}$	–	0	0

2.5 Proof of Proposition 2.13

Let $C = \Omega(n, q, q^m, L, \mathcal{P})$ be an nth-root code of length $N = |L|$. We have:

- in the zerofree case, $L = \{\alpha^{i_1}, \ldots, \alpha^{i_N}\} = \{\alpha^{i_j}\}_{i_j \in I}$, $I \subset \{1, \ldots, n\}$ such that $|I| = N$ and $i_1 < \ldots < i_N$, i.e. set I contains the exponents i_j such that α^{i_j} belongs to L.
- in the non-zerofree case, $L = \{\alpha^{i_1}, \ldots, \alpha^{i_{N-1}}, 0\} = \{\alpha^{i_j}\}_{i_j \in \hat{I}} \cup \{0\}$, $\hat{I} \subset \{1, \ldots, n\}$ such that $|\hat{I}| = N - 1$ and $i_1 < \ldots < i_{N-1}$, i.e. set $\hat{I}$ contains the exponents i_j such that α^{i_j} belongs to L.

Observe that having ordered the exponents of α in I (or $\hat{I}$) we have not ordered set L.

Let π be a projection map:

$$\pi : (\mathbb{F}_q)^{n+1} \to (\mathbb{F}_q)^N, \quad \pi : (v_1, \ldots, v_n, v_0) \mapsto (v_{i_1}, \ldots, v_{i_N})$$

Note that we use v_0 to denote the last position of the input vector instead of v_{n+1}, in order to simplify notation in the non-zerofree case. For any w, $2 \leq w \leq N$, let $\mathcal{A}_w \subset (\mathbb{F}_{q^m})^w \times (\mathbb{F}_q)^w$ be the set composed of all vectors $\mathfrak{a} = (a_1, \ldots, a_w, a'_1, \ldots, a'_w)$ such that: $a_i = 0$ or $a_i = \alpha^j$ $(j = 1, \ldots, n)$, if $a_i = 0$ then $i = w$, $a_i \neq a_j$ $(\forall i \neq j)$, and $a'_i \neq 0$ (for any i').

Sets $\{\mathcal{A}_w\}_{2 \leq w \leq N}$ are obviously disjoint. We define a function ϕ as

$$\phi : \bigsqcup_{2 \leq w \leq N} \mathcal{A}_w \to (\mathbb{F}_q)^{n+1}, \quad \phi(a_1, \ldots, a_w, a'_1, \ldots, a'_w) = (v_1, \ldots, v_n, v_0),$$

where

$$v_0 = \begin{cases} a'_w, \text{ if } a_w = 0 \\ 0, \quad \text{if } a_w \neq 0 \end{cases}, \text{ and for } i \neq 0, v_i = \begin{cases} a'_j, \text{ if } \exists j \text{ such that } a_j = \alpha^i \\ 0, \quad \text{otherwise} \end{cases}.$$

Composing maps ϕ and π we obtain $\Phi = \pi \circ \phi : \bigsqcup_{2 \leq w \leq N} \mathcal{A}_w \to (\mathbb{F}_q)^N$,

$$\pi \circ \phi(a_1, \ldots, a_w, a'_1, \ldots, a'_w) = \pi(v_1, \ldots, v_n, v_0) = (v_{i_1}, \ldots, v_{i_N}).$$

We claim that if $v \in (\mathbb{F}_q)^N$ is a vector of weight w then $\Phi^{-1}(v) \subset \mathcal{A}_w$ and

$$|\Phi^{-1}(v)| = \begin{cases} w! & \text{if } v \text{ is zerofree} \\ (w-1)! & \text{if } v \text{ is non-zerofree} \end{cases}. \tag{5}$$

In fact, let v be a vector of weight w:

$$\begin{array}{c} v = \Big(\underbrace{0,\ldots,0}, \ \nu_1, 0,\ldots,0, \nu_i, 0,\ldots,0, \nu_w, \underbrace{0,\cdots,0}\Big). \\ \qquad\uparrow \qquad\uparrow \qquad\qquad\uparrow \qquad\qquad\uparrow \qquad\uparrow \\ \mu_1-1 \quad \mu_1 \qquad\quad \mu_i \qquad\quad \mu_w \quad N-\mu_w \end{array} \tag{6}$$

If v is zerofree then any $\mathfrak{a} \in \Phi^{-1}(v)$ is $\mathfrak{a} = (\alpha^{\mu_1}, \ldots \alpha^{\mu_w}, \nu_1, \ldots, \nu_w)$, or any other vector obtained from $\mathfrak{a}$ via a permutation $\sigma \in \mathrm{S}_w$ acting as:

$$(\alpha^{\mu_{\sigma(1)}}, \ldots, \alpha^{\mu_{\sigma(w)}}, \nu_{\sigma(1)}, \ldots, \nu_{\sigma(w)}),$$

so that $|\Phi^{-1}(v)| = w!$.
If v is non-zerofree then any $\mathfrak{a} \in \Phi^{-1}(v)$ is $\mathfrak{a} = (\alpha^{\mu_1}, \ldots \alpha^{\mu_{w-1}}, 0, \nu_1, \ldots, \nu_{w-1}, \nu_w)$, or any other vector obtained from $\mathfrak{a}$ via a permutation $\tilde{\sigma} \in \mathrm{S}_{w-1}$ acting as:

$$(\alpha^{\mu_{\tilde{\sigma}(1)}}, \ldots, \alpha^{\mu_{\tilde{\sigma}(w-1)}}, 0, \nu_{\tilde{\sigma}(1)}, \ldots, \nu_{\tilde{\sigma}(w-1)}, \nu_w),$$

so that $|\Phi^{-1}(v)| = (w-1)!$ and (5) is proved.

Let $c = (c_1, \ldots, c_N) \in C$ be a codeword of weight w. Let H be the standard parity-check matrix of C, so that $Hc^T = 0$, i.e.

$$\begin{pmatrix} g_1(l_1) & g_1(l_2) & \ldots & g_1(l_N) \\ g_2(l_1) & g_2(l_2) & \ldots & g_2(l_N) \\ \vdots & \vdots & \vdots & \vdots \\ g_r(l_1) & g_r(l_2) & \ldots & g_r(l_N) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = 0.$$

By representing c as in (6) we obtain the r equations

$$\sum_{h=1}^{w} g_s(l_{\mu_h})\nu_h = 0 \qquad \text{for } s = 1, \ldots, r. \tag{7}$$

If c is zerofree, representing $l_{\mu_i} = \alpha^{\mu_i}$ with z_i and ν_i with y_i for any $i = 1, \ldots, w$, we define an ideal J'_w in $\mathbb{F}_{q^m}[z_1, \ldots, z_w, y_1, \ldots, y_w]$ as generated by $\{\sum_{h=1}^{w} y_h g_s(z_h)\}_{1 \le s \le r}$, $\{y_i^{q-1} - 1\}_{1 \le i \le w}$ and $\{z_i^n - 1\}_{1 \le i \le w}$.

If c is non-zerofree, representing $l_{\mu_i} = \alpha^{\mu_i}$ with z_i, $i = 1, \ldots, w-1$, and ν_i with y_i, $i = 1, \ldots, w$, we define an ideal $\hat{J}'_w$ in $\mathbb{F}_{q^m}[z_1, \ldots, z_w, y_1, \ldots, y_w]$ as generated by $\{\sum_{h=1}^{w} y_h g_s(z_h)\}_{1 \le s \le r}$, $\{y_i^{q-1} - 1\}_{1 \le i \le w}$, $\{z_i^n - 1\}_{1 \le i \le w-1}$, and z_w.

It is clear from equations (7) that any codeword of weight w corresponds to at least one solution of ideals J'_w, $\hat{J}'_w$: we will refer to these solutions as "proper solutions" and to the others as "spurious solutions".

We certainly have a spurious solution if, for $i \neq j$, we get $z_i = z_j$: a codeword cannot have for the same position different values. In order to remove these spurious solutions we add polynomials $\{p_{i,j}(z_i, z_j)\}_{\{1\leq i<j\leq w\}} = \frac{z_i^n - z_j^n}{z_i - z_j}$. In [Sal06] the author shows that the set $\{z_i^n - 1, p_{i,j}(z_i, z_j)\}_{\{1\leq i<j\leq w\}}$ is a basis for the ideal I vanishing on the variety

$$V = \{(\bar{z}_1, \ldots, \bar{z}_w) \mid \bar{z}_i^n - 1 = 0,\, i = 1, \ldots, w, \bar{z}_i \neq \bar{z}_j, 1 \leq i < j \leq w\}.$$

Then, we can add, respectively, to J'_w and $\hat{J}'_w$ sets $\{p_{i,j}(z_i, z_j)\}_{\{1\leq i<j\leq w\}}$ and $\{p_{i,j}(z_i, z_j)\}_{\{1\leq i<j\leq w-1\}}$, obtaining respectively

$$J''_w =< J'_w, p_{i,j}(z_i, z_j)_{\{1\leq i<j\leq w\}} > \text{ and } \hat{J}''_w =< \hat{J}'_w, p_{i,j}(z_i, z_j)_{\{1\leq i<j\leq w-1\}} > .$$

Observe that J''_w is J_w and $\hat{J}''_2$ has the same number of solutions of $\hat{J}_{w-1}$. In conclusion, for any word c of weight w, if c is zerofree there is at least one solution of J_w, else there is at least a solution of either J_w or $\hat{J}_{w-1}$.

Conversely, let $\check{c} = (\check{z}_1, \ldots, \check{z}_w, \check{y}_1, \ldots, \check{y}_w)$ be in the variety of $J''_w = J_w$. Since $\check{z}_i^n = 1$, $\forall i = 1, \ldots, w$, we can write any $\check{z}_i$ as α^{μ_i}, for some $1 \leq \mu_i \leq n$. Moreover, since $\check{z}_i \neq \check{z}_j$ for any $i \neq j$ we have that $\mu_i \neq \mu_j$ for any $i \neq j$ and $y_j \neq 0 \,\forall\, j = 1, \ldots, w$, so that $\check{c} \in \mathcal{A}_w$. Applying map Φ to $\check{c}$ we obtain:

$$c = \Big(\underbrace{0, \ldots, 0}_{\uparrow},\ \underset{\uparrow}{\check{y}_1},\ 0, \ldots, 0,\ \underset{\uparrow}{\check{y}_i},\ 0, \ldots, 0,\ \underset{\uparrow}{\check{y}_w},\ \underbrace{0, \cdots, 0}_{\uparrow}\Big).$$

$$\mu_1 - 1 \quad \mu_1 \qquad \mu_i \qquad \mu_w \quad N - \mu_w$$

A direct check shows that c is actually a codeword.
On the other hand, if $\check{c} = (\check{z}_1, \ldots, \check{z}_w, \check{y}_1, \ldots, \check{y}_w)$ is in the variety of $\hat{J}''_w = \hat{J}_{w-1}$, since $\check{z}_i^n = 1$, $\forall i = 1, \ldots, w-1$, we can write any $\check{z}_i$ as α^{μ_i}, for some $1 \leq \mu_i \leq n$. Moreover, since $\check{z}_i \neq \check{z}_j$ for any $i \neq j$, we have that $\mu_i \neq \mu_j$ for any $i \neq j$. We compute $\Phi(\check{c})$:

$$c = \Big(\underbrace{0, \ldots, 0}_{\uparrow},\ \underset{\uparrow}{\check{y}_1},\ 0, \ldots, 0,\ \underset{\uparrow}{\check{y}_i},\ 0, \ldots, 0,\ \underset{\uparrow}{\check{y}_{w-1}},\ 0, \ldots, 0,\ \underset{\uparrow}{\check{y}_w}\Big).$$

$$\mu_1 - 1 \quad \mu_1 \qquad \mu_i \qquad \mu_{w-1} \qquad \mu_w$$

A direct check shows that c is actually a codeword.

To conclude the proof it is enough to apply (5).

3 General error locator polynomial

In this section we assume the reader is familiar with Gröbner theory for 0−dimensional ideals, in particular with the Gianni-Kalkbrener theorem ([Kal89], [Gia89],[GM89],[CM02]).

Let $\mathbb{K}$ be a (not necessarily finite) field. Assume G is a Gröbner basis for a 0−dimensional ideal $J \subset \mathbb{K}[\mathcal{S}, \mathcal{A}, \mathcal{T}]$, $\mathcal{S} = (\mathsf{s}_1, \ldots, \mathsf{s}_H)$, $\mathcal{A} = (\mathsf{a}_1, \ldots, \mathsf{a}_L)$, $\mathcal{T} = (\mathsf{t}_1, \ldots, \mathsf{t}_M)$ w.r.t. a order with $\mathcal{S} < \mathcal{A} < \mathcal{T}$ and with the $\mathcal{A}$–variables lexicographically ordered by $\mathsf{a}_1 > \mathsf{a}_2 > \cdots > \mathsf{a}_L$. Then the elements of set $G \cap (\mathbb{K}[\mathcal{S}, \mathcal{A}] \setminus \mathbb{K}[\mathcal{S}])$ can be collected into blocks $\{G_i\}_{1 \leq i \leq L}$:

$$G_1 = \{g_{1,1}(\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_2, \mathsf{a}_1), \ldots, g_{1,l_1}(\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_2, \mathsf{a}_1)\},$$

$$G_2 = \{g_{2,1}(\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_2), \ldots, g_{2,l_2}(\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_2)\},$$

$$\vdots$$

$$G_L = \{g_{L,1}(\mathcal{S}, \mathsf{a}_L), \ldots, g_{L,l_L}(\mathcal{S}, \mathsf{a}_L)\},$$

in such a way that:

- for any i, $G_i \subset \mathbb{K}[\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_{i+1}][\mathsf{a}_i] \setminus \mathbb{K}[\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_{i+1}]$,
- the ideal generated by $\sqcup_{j>i} G_j$ is actually the i-th elimination ideal J_i, $J_i = J \cap \mathbb{K}[\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_i]$.

The Gianni-Kalkbrener Theorem ensures that $G_i \neq \emptyset$ for any $1 \leq i \leq L$. Clearly any G_i, $1 \leq i \leq L$, can be decomposed into blocks of polynomials according to their degree with respect to the variable a_i:

$$G_i = \cup_{\delta=1}^{\Delta_i} G_{i\delta},$$

but some $G_{i\delta}$ could be empty. In this way, if $g \in G_{i\delta}$, we have:

- $g \in \mathbb{K}[\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_{i+1}][\mathsf{a}_i] \setminus \mathbb{K}[\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_{i+1}]$,
- $\deg_{\mathsf{a}_i}(g) = \delta$, i.e. $g = b\mathsf{a}_i^\delta + \cdots$ and $b = \mathrm{Lp}(g) \in \mathbb{K}[\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_{i+1}]$.

Let $N_{i\delta}$ be the number of elements of $G_{i\delta}$. We name the elements of the set $G_{i\delta} = \{g_{i\delta j}, 1 \leq j \leq N_{i\delta}\}$ after their order:

$$h < j \Leftrightarrow \mathrm{Lt}(g_{i\delta h}) < \mathrm{Lt}(g_{i\delta j}).$$

Remark 3.1. We can summarize our description.
Given any two polynomials $g_{lDh} \in G_{lD}$ and $g_{i\delta j} \in G_{i\delta}$, then

$$g_{lDh} < g_{i\delta j} \Leftrightarrow \mathrm{Lt}(g_{lDh}) < \mathrm{Lt}(g_{i\delta j}) \Leftrightarrow \begin{cases} l > i \text{ or} \\ l = i, D < \delta \text{ or} \\ l = i, D = \delta, h < j \end{cases} \tag{8}$$

Since J is $0-$dimensional, we can clearly decompose the variety of its elimination ideals as follows. Let $J_{\mathcal{S}} = J\cap\mathbb{K}[\mathcal{S}]$, $J_{\mathcal{S}\cup\{\mathsf{a}_L\}} = J\cap\mathbb{K}[\mathcal{S},\mathsf{a}_L]$, $\ldots$, $J_{\mathcal{S}\cup\{\mathsf{a}_L,\ldots,\mathsf{a}_1\}} = J\cap\mathbb{K}[\mathcal{S},\mathsf{a}_L,\ldots,\mathsf{a}_1] = J\cap\mathbb{K}[\mathcal{S},\mathcal{A}]$. We have:

1) $\mathcal{V}(J_{\mathcal{S}}) = \sqcup_{j=1}^{\lambda(L)}\Sigma_j^L$, with

$$\Sigma_j^L = \{(\overline{\mathsf{s}}_1,\ldots,\overline{\mathsf{s}}_N)\in\mathcal{V}(J_{\mathcal{S}}) \mid \text{there are exactly } j \text{ values } \{\bar{\mathsf{a}}_L^{(1)},\ldots,\bar{\mathsf{a}}_L^{(j)}\},\ s.t.(\overline{\mathsf{s}}_1,\ldots,\overline{\mathsf{s}}_N,\bar{\mathsf{a}}_L^{(i)})\in\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_L\}}), 1\le i\le j\};$$

2) $\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_L\}}) = \sqcup_{j=1}^{\lambda(L-1)}\Sigma_j^{L-1}$,with

$$\Sigma_j^{L-1} = \{(\overline{\mathsf{s}}_1,\ldots,\overline{\mathsf{s}}_N,\overline{\mathsf{a}}_L)\in\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_L\}}) \mid \text{there are exactly } j \text{ values } \{\bar{\mathsf{a}}_{L-1}^{(1)},\ldots,\bar{\mathsf{a}}_{L-1}^{(j)}\}, s.t.(\overline{\mathsf{s}}_1,\ldots,\overline{\mathsf{s}}_N,\overline{\mathsf{a}}_L,\bar{\mathsf{a}}_{L-1}^{(i)})\in\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_L,\mathsf{a}_{L-1}\}}), 1\le i\le j\};$$

3) $\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_L,\ldots,\mathsf{a}_h\}}) = \sqcup_{j=1}^{\lambda(h-1)}\Sigma_j^{h-1}$, $2\le h\le L$ with

$$\Sigma_j^{h-1} = \{(\overline{\mathsf{s}}_1,\ldots,\overline{\mathsf{s}}_N,\overline{\mathsf{a}}_L,\ldots,\overline{\mathsf{a}}_h)\in\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_l,\ldots,\mathsf{a}_h\}}) \mid \exists \text{ exactly } j \text{ values } \{\bar{\mathsf{a}}_{h-1}^{(1)},\ldots,\bar{\mathsf{a}}_{h-1}^{(j)}\}, s.t.(\overline{\mathsf{s}}_1,\ldots,\overline{\mathsf{s}}_N,\overline{\mathsf{a}}_L,\ldots,\overline{\mathsf{a}}_h,\ \bar{\mathsf{a}}_{h-1}^{(i)})\in\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_L,\ldots,\mathsf{a}_{h+1}\}}), 1\le i\le j\};$$

Note that, for a general 0-dimensional ideal J, nothing can be said about $\lambda(h)$, except that $\lambda(h)\ge 1$ for any $2\le h\le L$.

We now introduce a class of ideals which are very useful in our context.

Definition 3.2. *With the above notation we say that J is* ***stratified*** *if:*

(1) $\lambda(h) = h$, $1\le h\le L$ *and*

(2) $\sum_j^h\neq\emptyset$, $1\le h\le L$, $1\le j\le h$.

In the next two examples we show some non-stratified ideals.

Example 3.3. Let $\mathcal{S} = \{\mathsf{s}_1\}$, $\mathcal{A} = \{\mathsf{a}_1\}$ (so that $L = 1$) and $\mathcal{T} = \{\mathsf{t}_1\}$ such that $\mathcal{S} < \mathcal{A} < \mathcal{T}$. Let $K = \mathbb{C}$ and J be the ideal in $\mathbb{C}[\mathsf{s}_1,\mathsf{a}_1,\mathsf{t}_1]$ generated by

$$\{\mathsf{s}_1^2-\mathsf{s}_1,\mathsf{a}_1\mathsf{s}_1-\mathsf{a}_1-\mathsf{s}_1+1,\mathsf{a}_1^2-2\mathsf{a}_1\mathsf{s}_1^2-2\mathsf{a}_1\mathsf{s}_1-\mathsf{a}_1+\mathsf{s}_1^3+3\mathsf{s}_1^2+2\mathsf{s}_1,\mathsf{t}_1\}.$$

The variety of J is $\mathcal{V}(J) = \{(0,1,0),(1,2,0),(1,3,0)\}$. Let $J_{\mathcal{S}} = J\cap\mathbb{C}[\mathcal{S}] = <\mathsf{s}_1^2-\mathsf{s}_1>$, then $\mathcal{V}(J_{\mathcal{S}}) = \sqcup_{j=1}^{\lambda(L)}\sum_j^L = \sqcup_{j=1}^{\lambda(1)}\sum_j^1 = \{0,1\}$. Clearly $\{0\} = \sum_1^1$ and $\{1\} = \sum_2^1$, which means $\lambda(1) = 2\neq 1 = L$, so ideal J is not stratified because condition (1) in Definition 3.2 is not satisfied for $h = 1$. See Figure 1 (A).

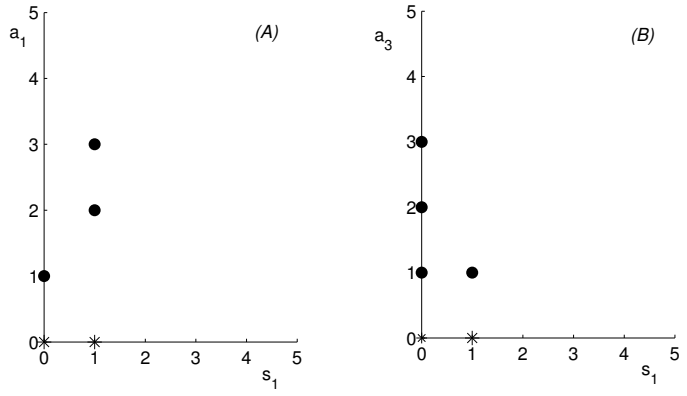

Fig. 1. Varieties in the non-stratified case.

Example 3.4. Let $\mathcal{S} = \{\mathsf{s}_1\}$, $\mathcal{A} = \{\mathsf{a}_1, \mathsf{a}_2, \mathsf{a}_3\}$ (so that $L = 3$) and $\mathcal{T} = \{\mathsf{t}_1\}$ such that $\mathcal{S} < \mathcal{A} < \mathcal{T}$ and $\mathsf{a}_1 > \mathsf{a}_2 > \mathsf{a}_3$. Let $K = \mathbb{C}$ and J be the ideal in $\mathbb{C}[\mathsf{s}_1, \mathsf{a}_1, \mathsf{a}_2, \mathsf{a}_3, \mathsf{t}_1]$ generated by

$$\{\mathsf{s}_1^2 - \mathsf{s}_1, \mathsf{a}_3\mathsf{s}_1 - \mathsf{s}_1,\ \mathsf{a}_3^3 - 6\mathsf{a}_3^2 + 11\mathsf{a}_3 - 6,\ \mathsf{a}_1, \mathsf{t}_1\}\,.$$

The variety of J is $\mathcal{V}(J) = \{(0,0,0,1,0), (0,0,0,2,0), (0,0,0,3,0), (1,0,0,1,0)\}$ and $\mathcal{V}(J_{\mathcal{S}\cup\mathsf{a}_3}) = \{(0,1), (0,2), (0,3), (1,1)\}$. Let $J_{\mathcal{S}} = J \cap \mathbb{C}[\mathcal{S}] =< \mathsf{s}_1^2 - \mathsf{s}_1 >$, then $\mathcal{V}(J_{\mathcal{S}}) = \sqcup_{j=1}^{\lambda(L)} \sum_j^L = \sqcup_{j=1}^{\lambda(3)} \sum_j^3 = \{0,1\}$. Clearly $\{1\} = \sum_1^3$ and $\{0\} = \sum_3^3$ which means $\lambda(3) = 3$, satisfying condition (1) in Definition 3.2. However, $\sum_2^3 = \emptyset$ and so ideal J is not stratified, because it does not satisfy condition (2) in Definition 3.2, for $h = 3$. See Figure 1 (B).

The next example shows a simple stratified ideal.

Example 3.5. Let $\mathcal{S} = \{\mathsf{s}_1\}$, $\mathcal{A} = \{\mathsf{a}_1, \mathsf{a}_2\}$ (so that $L = 2$) and $\mathcal{T} = \{\mathsf{t}_1\}$ such that $\mathcal{S} < \mathcal{A} < \mathcal{T}$ and $\mathsf{a}_1 > \mathsf{a}_2$. Let $K = \mathbb{C}$ and J be the ideal in $\mathbb{C}[\mathsf{s}_1, \mathsf{a}_1, \mathsf{a}_2, \mathsf{t}_1]$ generated by:

$$\{\mathsf{s}_1^2 - \mathsf{s}_1, \mathsf{a}_2 - 3, \mathsf{a}_1\mathsf{s}_1 - 2\mathsf{s}_1, \mathsf{a}_1^2 + \mathsf{a}_1\mathsf{s}_1 - 3\mathsf{a}_1 - 2\mathsf{s}_1 + 2, \mathsf{t}_1\}.$$

The variety of J is $\mathcal{V}(J) = \{(0,1,3,0), (0,2,3,0), (1,2,3,0)\}$. Let $J_{\mathcal{S}} = J \cap \mathbb{C}[\mathcal{S}] =< \mathsf{s}_1^2 - \mathsf{s}_1 >$, then $\mathcal{V}(J_{\mathcal{S}}) = J_{\mathcal{S}} = \sqcup_{j=1}^{\lambda(L)} \sum_j^L = \sqcup_{j=1}^{\lambda(2)} \sum_j^2 = \{0,1\}$. Clearly $\{1\} = \sum_1^2$ and $\{0\} = \sum_2^2$, which means $\lambda(2) = 2$ satisfying condition (1) in Definition 3.2, for $h = 1, 2$. Variety $\mathcal{V}(J_{\mathcal{S}\cup\{\mathsf{a}_2\}}) = \sqcup_{j=1}^{\lambda(L-1)} \sum_j^{\lambda(1)} = \{(0,1), (0,2), (1,2)\}$. On the other hand, $\{(0,1), (0,2), (1,2)\} = \sum_1^1$, which means $\lambda(L-1) = \lambda(1) = 1$ satisfying condition (1) and all $\sum_j^i$, $\forall i, j = 1, 2$, are not empty, so that ideal J is stratified. See Figure 2 (A) and (B).

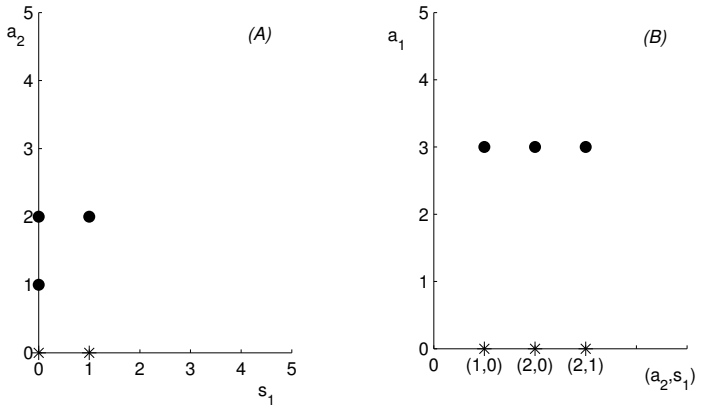

Fig. 2. Varieties in a stratified case

.

By revisiting Theorem 5.4 and Theorem 5.5 of [OS05], we get the following theorem.

Theorem 3.6. *Let J be a radical, stratified ideal, then for $1 \leq i \leq L$,*

$$G_i = \sqcup_{\delta=1}^{i} G_{i\delta} \ ,$$

with $G_{i\delta} \neq \emptyset$, $1 \leq \delta \leq i$ and $1 \leq i \leq L$. Moreover

- *$\forall\, 1 \leq i \leq L$, $G_{ii} = \{g_{ii1}\}$, i.e. only one polynomial exists in G_i with degree i w.r.t a_i;*
- *$\forall\, 1 \leq i \leq L$, $\mathrm{Lp}(g_{ii1}) = 1$, $\mathrm{Lt}(g_{ii1}) = \mathsf{a}_i^i$.*

In next definition we adapt ideal J_w in Definition 2.10 to correct errors.

Definition 3.7. *Let $C = \Omega(q, n, q^m, L, \mathcal{P})$ be a zerofree, maximal nth-root code, with correction capability t. We denote by $J^{C,t}$ the ideal*
$J^{C,t} \subset \mathbb{F}_{q^m}[x_1, \ldots, x_r, z_t, \ldots, z_1, y_1, \ldots, y_t]$,

$$\begin{aligned} J^{C,t} = \langle \quad & \left\{\textstyle\sum_{h=1}^{t} y_h g_s(z_h) - x_s\right\}_{1\leq s\leq r}, \left\{y_j^{q-1} - 1\right\}_{1\leq j\leq t}, \\ & \left\{z_i z_j p_{ij}(z_i, z_j)\right\}_{i\neq j,\, 1\leq i,j\leq t}, \left\{z_j^n - z_j\right\}_{1\leq j\leq t} \quad \rangle \end{aligned} \tag{9}$$

We denote by $\mathcal{G}^{c,t}$ any Gröbner basis of $J^{C,t}$ w.r.t. $>$.

Note that variables $x_1, \ldots, x_r$ represent correctable syndromes, $z_1, \ldots, z_t$ error locations and $y_1, \ldots, y_t$ error values.

Lemma 3.8. *Ideal $J^{C,t}$ is radical and stratified.*

Proof. Points in $\mathcal{V}(J^{C,t})$ cannot be outside $(\mathbb{F}_{q^m})^{r+2t}$. Since $J^{C,t}$ contains polynomials $y_j^{q-1}-1$, and z_j^n-1 divides $z_j^{q^m}-z_j$ for $j=1,\ldots,t$, thanks to Seidenberg's Lemma ([Sei74]), $J^{C,t}$ is radical.

To prove that $J^{C,t}$ is stratified we begin with the case $h=t$ $(L=t)$.

Let μ be a natural number $2\leq\mu\leq t-1$ and

$$\Sigma_\mu^t=\{\bar{x}=(\bar{x}_1,\ldots\bar{x}_r)\in\mathcal{V}(J^{C,t}_{x_1,\ldots,x_r})\mid \text{ there are exactly}$$

$$\mu \text{ values } (z_t^1,\ldots,z_t^\mu) \text{ s.t. } (\bar{x}_1,\ldots\bar{x}_r,z_t^j)\in\mathcal{V}(J^{C,t}_{x_1,\ldots,x_r,z_t})\}.$$

A point in $\mathcal{V}(J^{C,t})$ corresponding to a syndrome that can correct $\mu-1$ errors is of type

$$(\bar{x}_1,\ldots,\bar{x}_r,\quad \underbrace{*\,*\,*\,*\,*}_{\mu-1 \text{ values } \neq 0},\quad \bar{y}_1,\ldots,\bar{y}_t).$$

$$\text{and } t-\mu+1 \text{ values } 0$$

There are $(\mu-1)!$ points corresponding to this syndrome. If we truncate them at the $(r+1)$-th component, position $r+1$ may assume either $\mu-1$ values corresponding to error locations or a zero value, for a total number of μ values. So

$$\Sigma_\mu^t\supset\{\text{syndromes that can correct } \mu-1 \text{ errors },2\leq\mu\leq t-1\}.$$

The converse inclusion is proved similarly.

Let

$$\Sigma_t^t=\{\bar{x}=(\bar{x}_1,\ldots\bar{x}_r)\in\mathcal{V}(J^{C,t}_{x_1,\ldots,x_r})\mid \text{ there are exactly}$$

$$t \text{ values } (z_t^1,\ldots,z_t^t) \text{ s.t. } (\bar{x}_1,\ldots\bar{x}_r,z_t^j)\in\mathcal{V}(J^{C,t}_{x_1,\ldots,x_r,z_t})\}.$$

As in the previous case, there may be points corresponding to syndromes correcting $t-1$ errors (in the $r+1$-th position we can find the $t-1$ values corresponding to errors positions and also 0), but there can also be points corresponding to syndromes correcting t errors, so that in position $r+1$ only nonzero values can stay (which are t).

Let

$$\Sigma_1^t=\{\bar{x}=(\bar{x}_1,\ldots\bar{x}_r)\in\mathcal{V}(J^{C,t}_{x_1,\ldots,x_r})\mid \text{ there is exactly}$$

$$\text{one value } z_t^1 \text{ s.t. } (\bar{x}_1,\ldots\bar{x}_r,z_t^1)\in\mathcal{V}(J^{C,t}_{x_1,\ldots,x_r,z_t})\}$$

and the only vector satisfying this condition is clearly vector $\underline{0}$ that can be extended only with a zero.

If $\mu>t$

$$\Sigma_\mu^t=\{\bar{x}=(\bar{x}_1,\ldots\bar{x}_r)\in\mathcal{V}(J^{C,t}_{x_1,\ldots,x_r})\mid \text{ there are exactly}$$

$$\mu \text{ values } (z_t^1, \ldots, z_t^t) \text{ s.t. } (\bar{x}_1, \ldots \bar{x}_r, z_t^j) \in \mathcal{V}(J^{C,t}_{x_1,\ldots,x_r,z_t})\}.$$

The syndromes can correct only $\mu \leq t$ errors, so that $\Sigma_\mu^t = \emptyset$ for all $\mu > t$.

We have proved that ideal $J^{C,t}$ for $h = t$ satisfies conditions (1) and (2) in Definition 3.2. With similar arguments we can prove that it satisfies these conditions for $h \neq t$, and hence it is stratified. □

Applying Theorem 3.6 to $J^{C,t}$, thanks to Lemma 3.8, we have the following fact.

Fact 3.9. *In Gröbner basis $\mathcal{G}^{C,t}$ there exists a unique polynomial of type*

$$g = z_t^t + \mathsf{a}_{t-1} z_t^{t-1} + \ldots + \mathsf{a}_0, \mathsf{a}_i \in \mathbb{F}_{q^m}[X].$$

Proof. It is enough to take $i = t$ and $g = g_{tt1}$. □

We are ready for the main result of this section.

Theorem 3.10. *If code C is a proper maximal zerofree nth-root code with correction capability t, then C possesses a general error locator polynomial.*

Proof. From Fact 3.9, a polynomial of type $g = z_t^t + \mathsf{a}_{t-1} z_t^{t-1} + \ldots + \mathsf{a}_0$, with $\mathsf{a}_i \in \mathbb{F}_{q^m}[X]$, exists in $J^{C,t}$. Since C is proper, all polynomials in ideal $J^{C,t}$ have coefficients in $\mathbb{F}_q$ and so g must be in $\mathbb{F}_q[X, z_t]$.
We claim that $\mathcal{L} = g(X, z_t) \in \mathbb{F}_q[X, z_t]$ is a general error locator polynomial for C. Polynomial g satisfies clearly (1) in Definition 1.9. Condition (2) in Definition 1.9 is satisfied, because correctable syndromes are in $\mathcal{V}(J^{C,t} \cap \mathbb{F}_q[X])$ and g is in $J^{C,t}$. □

Since cyclic codes are proper maximal zerofree nth-root codes (see Subsection 4.1) we obtain, as a special case of Theorem 3.10, that cyclic codes have general error locator polynomials (Theorem 6.9 in [OS05]).

In the next two examples we show two methods to compute general error locator polynomials. The former is suggested by Lemma 3.9. In the latter we assume we know that a general locator polynomial exists for the code and hence we apply directly Definition 1.9.

Example 3.11. Let G and H be the following binary matrices

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

Let C be the $[5, 2, 3]$ linear code over $\mathbb{F}_2$ with G as a generator matrix and H as a parity-check matrix. Note $t = 1$. Let γ be the primitive element of $\mathbb{F}_{16}$.

Then C is the zerofree maximal nth-root code $\Omega(2, 5, 2^4, R_5, \mathcal{P})$, where

$$\begin{aligned}\mathcal{P} = \{\ & g_1(x) = \gamma^4x^4 + \gamma^8x^3 + \gamma^2x^2 + \gamma x + 1,\\ & g_2(x) = \gamma^{10}x^4 + \gamma^5x^3 + \gamma^5x^2 + \gamma^{10}x + 1,\\ & g_3(x) = \gamma^{11}x^4 + \gamma^7x^3 + \gamma^{13}x^2 + \gamma^{14}x\}.\end{aligned}$$

We construct ideal $J^{C,t} \subset \mathbb{F}_{16}[x_1, x_2, x_3, z_1] = \mathbb{F}_{16}[X, Z]$, as follows:

$$J^{C,1} = \langle \{g_h(z_1) - x_h\}_{1\le h\le 3}, z_1^n - z_1\rangle.$$

If we calculate Gröbner basis $\mathcal{G}^{C,t} = \mathcal{G}_X \cup \mathcal{G}_{X,z_1}$ w.r.t. the lexicographical order induced by $x_1 < x_2 < x_3 < z_1$, we obtain:

$$\mathcal{G}_X = \{x_3^2 + x_3,\ x_2^2 + x_2,\ x_1x_3 + x_2x_3,\ x_1x_2 + x_1 + x_2x_3 + x_2 + x_3 + 1,\ x_1^2 + x_1\}$$

and

$$\mathcal{G}_{X,z_1} = \{g_{111} = \mathbf{z_1} + (\gamma^2{+}\gamma)x_1 + (\gamma^3{+}\gamma)x_2x_3 + \gamma x_2 + x_3 + (\gamma^3{+}\gamma^2{+}\gamma)\}.$$

In $\mathcal{G}_{X,z_1}$ there is only one polynomial in z_1 of degree 1, as we expected, g_{111}, and it must be a general error locator polynomial for C thanks to Fact 3.9.

Example 3.12. Let C be the code in Example 3.11. Another way to compute the general error locator polynomial is to see code C with parity-check matrix $H = (\gamma^6, \gamma^2, \gamma^3, \gamma^{14}, 1)$, so that $C = \Omega(2, 5, 2^4, R_5, \mathcal{P}')$, where

$$\mathcal{P}' = \{\gamma^{12}x^4 + \gamma^{11}x^3 + x^2 + \gamma^{14}x + \gamma^3\}.$$

If we calculate the Gröbner basis $\mathcal{G}'$ w.r.t. the lexicographical order induced by $x_1 < z_1$, its elements are:

$$\mathcal{G}'_{x_1} = x_1^5 + (\gamma^3)x_1^4 + (\gamma^3 + \gamma)x_1^2 + \gamma^2x_1 + (\gamma^2 + \gamma + 1),\ \mathcal{G}'_{x_1,z_1} = \mathbf{z_1} + x_1^3.$$

There is only one polynomial in z_1 of degree 1, as we expected, and it is another general error locator polynomial for C.

Example 3.13. Another way to compute general error locator polynomials for a code is to suppose that those polynomials exist. Let C be the code studied in Example 3.11. We assume that its parity-check matrix is a row, $H = (e_1, e_2, e_3, e_4, e_5)$. We search a general error locator polynomial $z + f(x)$ (the degree t of z is 1). It must satisfy the following conditions:

$$f(e_i) = \alpha^i, \quad \forall 1 \le i \le 5, \qquad \text{and } f(0) = 0.$$

Polynomial $f(x)$ has degree at most 5 with coefficients b_i in $\mathbb{F}_2$, so that we can write $f(x) = b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x$ $(f(0) = 0 \implies b_0 = 0)$.

We compute a Gröbner basis of ideal $J \subset \mathbb{F}_2[b_1, b_2, b_3, b_4, b_5, e_1, e_2, e_3, e_4, e_5]$,

$$J = \langle\ e_1 + e_2 + e_3,\ e_3 + e_4 + e_5,\ \{e_i^{15} + 1\}_{1 \le i \le 5},\ \{b_i^2 + b_i\}_{1 \le i \le 5},$$
$$f(e_1) + \gamma^3,\ f(e_2) + \gamma^6,\ f(e_3) + \gamma^9,\ f(e_4) + \gamma^{12},\ f(e_5) + \gamma^{15} \rangle\,,$$

where relations $e_1 = e_2 + e_3$, $e_4 = e_3 + e_5$ follow from matrix G. We obtain $e_1 = \gamma^6$, $e_2 = \gamma^2$, $e_3 = \gamma^3$, $e_4 = \gamma^{14}$, $e_5 = 1$, so that the parity-check matrix is $H = (\gamma^6, \gamma^2, \gamma^3, \gamma^{14}, 1)$ and the general error locator polynomial is $f(x) = x^3$. We note that it is the same as in Example 3.12.

Remark 3.14. The previous example is interesting because we have simultaneously computed for C an nth-root presentation and a general error locator polynomial. The nice shape of the general error locator polynomial reveals an *unexpected* structure in this code.

If the approach presented in Example 3.13 fails for a code C', that is, if $\mathcal{V}(J) = \emptyset$, then it means that C' does not possess a general error locator polynomial for any nth-root presentation, such that H is composed of one row. However, it could be that C' possesses a general error locator polynomial for H with up to $N-k$ rows. We think that it is obvious how this may be checked with a similar commutative algebra approach, and so we do not detail it.

3.1 Extended syndrome variety

We extend previous results to the case when there are also erasures. Let τ be a natural number corresponding to number of error, μ be a natural number corresponding to number of erasure and such that $2\tau + \mu < d$. We have to find solutions of equations of type:

$$\bar{s}_j + \sum_{l=1}^{\tau} a_l g_j(\alpha^{k_l}) + \sum_{\bar{l}=1}^{\nu} \bar{c}_{\bar{l}} g_j(\alpha^{h_{\bar{l}}}) = 0, \quad j = 1, \ldots, r\,, \tag{10}$$

where $\{k_l\}, \{a_l\}$ and $\{c_{\bar{l}}\}$ are unknown and $\{\bar{s}_j\}, \{h_{\bar{l}}\}$ are known. We introduce variables $W = (w_1, \ldots, w_\nu)$ and $U = (u_1, \ldots, u_\nu)$, where the $\{w_{\mathsf{h}}\}$ stand for erasure locations $(\alpha^{h_{\bar{l}}})$ and the $\{u_{\mathsf{h}}\}$ stand for erasure values $\bar{c}_{\bar{l}}$ $(\mathsf{h} = 1, \ldots, \nu)$.

When the word $v(x)$ is received, the number ν of erasures and their positions $\{w_{\mathsf{h}}\}$ are known.

We rewrite equations (10) in terms of X, Y, Z, W and U, where the $\{x_j\}$ stand for the syndromes $(j = 1, \ldots, r)$, as:

$$
\begin{aligned}
J^{C,\tau,\nu} = \langle \quad & \{\textstyle\sum_{l=1}^{\tau} y_l g_j(z_l) + \sum_{\bar{l}}^{\nu} u_{\bar{l}} g_j(w_{\bar{l}}) - x_j\}_{j=1,\ldots,r,} \\
& \{z_i^{n+1} - z_i\}_{i=1,\ldots,\tau}, && \{y_i^{q-1} - 1\}_{i=1,\ldots,\tau}, \\
& \{u_h^q - u_h\}_{h=1,\ldots,\nu}, && \{w_h^n - 1\}_{h=1,\ldots,\nu}, \\
& \{x_j^{q^m} - x_j\}_{j=1,\ldots,r}, && \{p(w_h, w_k)\}_{h\neq k,h,k=1,\ldots,\nu}, \\
& \{z_i p(z_i, w_h)\}_{i=1,\ldots,\tau,h=1,\ldots,\nu}, && \{z_i z_j p(z_i, z_j)\}_{i\neq j,i,j=1,\ldots,\tau}\rangle.
\end{aligned}
$$

We observe that:

- polynomials $\sum_{l=1}^{\tau} y_l g_j(z_l) + \sum_{\bar{l}}^{\nu} u_{\bar{l}} g_j(w_{\bar{l}}) - x_j$ characterize the nth-root code;
- polynomials $z_i^{n+1} - z_i$ ensure that z_i are nth-roots of unity or 0;
- polynomials $w_h^n - 1$ ensure that w_h are nth-roots of unity;
- polynomials $y_i^{q-1} - 1$, $u_h^q - u_h$ ensure that $y_i \in \mathbb{F}_q^*$ and $u_h \in \mathbb{F}_q$;
- polynomials $z_i p(z_i, w_h)$ ensure that an error cannot occur in a position corresponding to an erasure;
- polynomials $p(w_h, w_k)$ ensure that any two erasure locations are distinct;
- polynomials $z_i z_j p(z_i, z_j)$ ensure that any two error locations are distinct.

Ideal $J^{C,\tau,\nu}$ depends only on code C and on ν. With arguments similar to those used in the proof of Lemma 3.8 it is easy to show the following lemma:

Lemma 3.15. *Ideal $J^{C,\tau,\nu}$ is stratified and radical.*

Applying Theorem 3.6, thanks to Lemma 3.15, we get the following results:

Fact 3.16. *In Gröbner basis $\mathcal{G}^{C,\tau,\nu}$ there is a unique polynomial of type*

$$g = z_\tau^\tau + \mathsf{a}_{\tau-1} z^{\tau-1} + \ldots + \mathsf{a}_0,\ \mathsf{a}_i \in \mathbb{F}_{q^m}[X, W].$$

Theorem 3.17. *If code C is a proper maximal zerofree nth-root code, then C possesses general error locator polynomials of type ν, for any $\nu \geq 0$.*

Proof. It is enough to take g as in Fact 3.16. □

Example 3.18. Let C' be the shortened code obtained from code C presented in Example 2.4. Code C' is a $[7, 1, 6]$ linear code, so that τ (errors) and μ (erausers) satisfy relation $\tau + e < 6$. If $\tau = 1, e = 2$, the syndrome ideal is

$$
\begin{aligned}
J = \{ & g_1(z_1) + u_1 g_1(w_1) + u_2 g(w_2) + x_1, g_2(z_1) + u_1 g_2(w_1) + u_2 g_2(w_2) + x_2, \\
& z_1^8 - z_1, w_1^7 - 1, w_2^7 - 1, x_1^8 - x_1, x_2^8 + x_2, u_1^2 + u_1, u_2^2 + u_2, \\
& z_1 p(z_1, w_1, 7), z_1 p(z_1, w_2, 7), p(w_1, w_2, 7)\}
\end{aligned}
$$

and in G there is only one polynomial having z_1 as leading term (App. A).

4 Other code families

In this section we analyze some classes of codes and we show how they can be seen naturally as nth-root codes.

4.1 Cyclic codes and related codes

Definition 4.1. *Let g be a divisor of $x^n - 1$ over $\mathbb{F}_q$. We define S_C as the set*

$$S_C = \{i_1, \ldots, i_{n-k} | g(\alpha^{i_j}) = 0, 1 \leq i_j \leq n\}$$

of all powers of α that are roots of g. Let H be the following matrix:

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

The **cyclic** *code C generated by g is the linear code C over $\mathbb{F}_q$ such that H is a parity-check matrix for C.*

Setting q and n as above, m the smallest integer such that $n|q^m - 1$, $L = R_n$, i.e. $L = \{\alpha, \alpha^2, \ldots, \alpha^n\}$, and $\mathcal{P} = \{x^{i_j} \mid i_j \in S_C\}$, we can see C as the nth-root code $\Omega(q, n, q^m, R_n, \{x^{i_j} \mid i_j \in S_C\})$. In fact, nth-root codes are a generalization of cyclic codes. Moreover, since $x^h \in \mathbb{F}_q[x]$ for any q, we have the following result.

Proposition 4.2. *Any cyclic code is a proper maximal zerofree nth-root code. As a consequence, it possesses a general error locator polynomial.*

We claim that also shortened cyclic codes (see Definition 1.8) can be seen as nth-root codes: if D is a subset of positions where cyclic code C is shortened, then code $C(D)$ is an nth-root code $\Omega(q, n, q^m, L, \mathcal{P})$, where q, n and $\mathcal{P}$ are as above and $L = \{\alpha^j \mid 1 \leq j \leq n,\ j \notin D\}$.

Remark 4.3. Since shortened (and non-shortened) cyclic codes are nth-root codes, we can apply the algorithm of Subsection 2.3 to compute their distance and weight distribution. In this special case, this algorithm coincides with the algorithm proposed in [Sal06].

Now we consider the Reed-Solomon codes and the BCH codes, which are important families of cyclic codes.

Definition 4.4. *A cyclic code C of length n over $\mathbb{F}_q$ is a* **BCH** *code of designed distance δ if, for some integer $b \geq 0$, the generator polynomial $g(x)$ of C is the monic lowest degree polynomial over $\mathbb{F}_q$ having $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}$ as zeros.*

The minimum distance is $d \geq \delta$ and the parity-check matrix is:

$$H = \begin{pmatrix} 1 & \alpha^b & \alpha^{2b} & \dots & \alpha^{n-1} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \dots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+\delta-2} & \alpha^{2(b+\delta-2)} & \dots & \alpha^{(n-1)(b+\delta-2)} \end{pmatrix}$$

Definition 4.5. *A Reed-Solomon (or* **RS***) code over $\mathbb{F}_q$ is a BCH code of length $N = q - 1$.*

Usually, but not always, $b = 1$. A RS code is a cyclic code with generator polynomial $g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \dots (x - \alpha^{b-\delta-2})$, where α is the primitive element of $\mathbb{F}_{q^m}$. A RS code can be treated as an nth-root code $\Omega(q, n, q^m, \mathbb{F}^*_{q^m}, \{x^i \mid i = b, b+1, \dots, b+\delta-2\})$.
Remark 4.6. Using result from [KM00], it is easy to describe explicitly a general error locator polynomial for RS codes and hence prove its high sparsity.

4.2 Classical Goppa codes

In this section we view classical Goppa codes as nth-root codes.

Definition 4.7. *Let $g(z) \in \mathbb{F}_{q^m}[z]$, $\deg(g) = r \geq 2$, and let $L = \{\alpha_1, \dots, \alpha_N\}$ denote a subset of elements of $\mathbb{F}_{q^m}$ which are not roots of $g(z)$. Then the* **Goppa** *code $\Gamma(L, g)$ is defined as the set of all vectors $c = (c_1, \dots, c_N)$ with components in $\mathbb{F}_q$ that satisfy the condition:*

$$\sum_{i=1}^{N} \frac{c_i}{z - \alpha_i} \equiv 0 \mod g(z) .$$

Usually, but now always, set L is taken to be the set of all elements in $\mathbb{F}_{q^m}$ which are not roots of the Goppa polynomial $g(z)$. If $g(z)$ is irreducible over $\mathbb{F}_{q^m}$ then code $\Gamma(L, g)$ is called *irreducible Goppa code*. A parity-check matrix for $\Gamma(L, g)$ can be written as:

$$\begin{pmatrix} \frac{1}{g(\alpha_1)} & \frac{1}{g(\alpha_2)} & \dots & \frac{1}{g(\alpha_N)} \\ \frac{\alpha_1}{g(\alpha_1)} & \frac{\alpha_2}{g(\alpha_2)} & \dots & \frac{\alpha_N}{g(\alpha_N)} \\ \frac{\alpha_1^2}{g(\alpha_1)} & \frac{\alpha_2^2}{g(\alpha_2)} & \dots & \frac{\alpha_N^2}{g(\alpha_N)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_1^{r-1}}{g(\alpha_1)} & \frac{\alpha_2^{r-1}}{g(\alpha_2)} & \dots & \frac{\alpha_N^{r-1}}{g(\alpha_N)} \end{pmatrix} .$$

Setting q, m and L as above, $n = q^m - 1$ and $\mathcal{P} = \{\frac{x^i}{g(x)}, \forall i = 0, \dots, r-1\}$, it

follows that classical Goppa code $\Gamma(L, g)$ over $\mathbb{F}_q$ is the nth-root code

$$\Gamma = \Omega\left(q, q^m - 1, q^m, L, \left\{\frac{x^i}{g(x)} | i = 0, \ldots, r-1\right\}\right).$$

The following results are then obvious.

Proposition 4.8. *If the Goppa polynomial g is in $\mathbb{F}_q[x]$, then $\Gamma(L, g)$ is a proper nth-root code. In particular, if $L = \mathbb{F}_{q^m} \setminus \{0\}$, code $\Gamma(L, g)$ is proper and maximal.*

Theorem 4.9. *Any classical Goppa code $\Gamma(L, g)$ such that $g \in \mathbb{F}_q[x]$ and $L = \mathbb{F}_{q^m} \setminus \{0\}$ admits a general error locator polynomial.*

Example 4.10. Consider the nth-root code of Example 2.4, shortened in position 0. It is a classical Goppa code with $g(x) = x^2 + x + 1$ and $L = \mathbb{F}_8^*$. A general error locator polynomial for this code is

$$\begin{aligned}\mathcal{L} =& \mathbf{z_2^2} + z_2(x_1^5x_2^2 + x_1^5 + x_1^3x_2^2 + x_1^3 + x_1^2x_2^2 + x_1^2x_2 + x_1x_2^5 + x_1x_2^4 + x_1x_2^3 + \\ & x_1x_2^2 + x_1x_2 + x_1 + x_2^7 + x_2^4 + x_2^3 + x_2^2 + 1) + x_1^5x_2^2 + x_1^5x_2 + x_1^5 + x_1^4x_2^2 + \\ & x_1^3x_2^3 + x_1^2x_2 + x_1^2 + x_1x_2^6 + x_1x_2 + x_1 + x_2^7 + x_2^6\,.\end{aligned}$$

Now we focus on irreducible Goppa codes, $\Gamma(L, g)$ such that $L = \mathbb{F}_{q^m}$. These codes admit also the following parity-check matrix H:

$$H = \left(\tfrac{1}{\gamma - \zeta_0}, \tfrac{1}{\gamma - \zeta_1}, \cdots, \tfrac{1}{\gamma - \zeta_{q^m-1}}\right),$$

where $\gamma \in \mathbb{F}_{q^{mr}}$ is any root of $g(x)$ and $\mathbb{F}_{q^m} = \{\zeta_i \mid 0 \le i \le q^m - 1\}$.
We can extend Definition 2.1 to *generalized* nth-root codes, by allowing also $\mathcal{P} \subset \mathbb{F}_Q[X]$ with $\mathbb{F}_{q^m} \subset \mathbb{F}_Q$. In this sense, an irreducible Goppa code $\Gamma(L, g)$ can be considered as a generalized nth-root code $\Omega(q, q^m - 1, q^{mr}, \mathbb{F}_{q^{mr}}, \mathcal{P})$, where $\mathcal{P} = \{g(x)\} = \{\frac{1}{\gamma - x}\}$.
Even the ideals in Definition 2.10 can be given in the generalized case, by considering $J_w = J_w(C) = J_w(q, n, q^{mr}, L, \mathcal{P}) \subset \mathbb{F}_Q[z_1, \ldots, z_w, y_1, \ldots, y_w]$, $\hat{J}_w = \hat{J}_w(C) = \hat{J}_w(q, n, q^{mr}, L, \mathcal{P}) \subset \mathbb{F}_Q[z_1, \ldots, z_w, y_1, \ldots, y_w, \mu]$.

Example 4.11. Let us consider the nth-root code C given in Example 2.4. Polynomial $g(x)$ is irreducible over $\mathbb{F}_{2^3} = \{\zeta_i\}_{i=0,\ldots,7}$, so Goppa code C is irreducible. Let ε be a primitive element of $\mathbb{F}_{64}$: $\gamma = \varepsilon^{21}$ is a root of Goppa polynomial $g(x)$ and $f(x) = x^6 + x^4 + x^3 + x + 1$ is a primitive polynomial of $\mathbb{F}_{64}$ over $\mathbb{F}_2[x]$. Parity-check matrix H is then:

$$H = \left(\tfrac{1}{\varepsilon^{21} - \zeta_0}, \tfrac{1}{\varepsilon^{21} - \zeta_1}, \ldots, \tfrac{1}{\varepsilon^{21} - \zeta_7}\right).$$

Setting q, q^m and $L = \mathbb{F}_{q^m}$ as above, $n = q^m - 1$ and $\mathcal{P} = \{g(x)\} = \{\frac{1}{\varepsilon^{21} - x}\}$, we can see C as a generalized nth-root code.

4.3 Reed-Muller codes

Definition 4.12. *Let $m \in \mathbb{N}$, $m \geq 1$. An arbitrary function $f : (\mathbb{F}_2)^m \longrightarrow \mathbb{F}_2$ is called a* **Boolean function** *(***B.f.** *for short).*

For example we can define the i-th **elementary** B.f., $v_i : (\mathbb{F}_2)^m \longrightarrow \mathbb{F}_2$, $v_i(x_1, \ldots, x_m) = x_i$. Their products form a linear basis for all the B.f.'s

$$\{1, v_1, \ldots, v_m, v_1v_2, v_1v_3, \ldots, v_{m-1}v_m, v_1v_2v_3, \ldots, v_1v_2\cdots v_m\},$$

so that we can see the B.f's. as polynomials in $\mathbb{F}_2[v_1, v_2, \ldots, v_m]$ ([MS77]).

Definition 4.13. *Let $m \geq 1$ and $1 \leq r \leq m$. We define the* **binary Reed-Muller code** *of order r and length $n = 2^m$ as the set of Boolean functions that are polynomials of degree at most r. We denote this set by* $\mathbf{RM(r, m)}$.

The key point is that we can associate to any B.f. f a vector $\underline{f}$ such that $\underline{f} = (f(V_1), \ldots, f(V_{2^m}))$, once an ordering on $(\mathbb{F}_2)^m = \{V_i\}_{1 \leq i \leq 2^m}$ has been chosen (which we assume in this sub-section).

Theorem 4.14 ([MS77])**.** *The dual code of $RM(r, m)$ is $RM(m - r, m)$.*

Hence we can construct a linear basis for the dual code of $RM(r, m)$ by taking a linear basis for $RM(m - r, m)$, so that a parity-check matrix is

$$H = \begin{pmatrix} \underline{v_1} \\ \vdots \\ \underline{v_m} \\ \underline{v_1v_2} \\ \vdots \\ \underline{v_{m-r+1}\cdots v_m} \end{pmatrix} =$$

$$\begin{pmatrix} v_1(V_1) & v_1(V_2) & \ldots & v_1(V_{2^m}) \\ \vdots & \vdots & \vdots & \vdots \\ v_m(V_1) & v_m(V_2) & \ldots & v_m(V_{2^m}) \\ (v_1v_2)(V_1) & (v_1v_2)(V_2) & \cdots & (v_1v_2)(V_{2^m}) \\ \vdots & \vdots & \vdots & \vdots \\ (v_{m-r+1}\cdots v_m)(V_1) & (v_{m-r+1}\cdots v_m)(V_2) & \ldots & (v_{m-r+1}\cdots v_m)(V_{2^m}) \end{pmatrix}.$$

In other words, code $RM(r, m)$ can be seen as the nth-root code

$$\Omega(2, 2^m - 1, 2^m, \mathbb{F}_{2^m}, \{v_{i_1}\cdots v_{i_j} | 1 \leq j \leq m - r,\ 1 \leq i_1 \neq \ldots \neq i_j \leq m\}).$$

4.4 Algebraic-geometry codes

Let $S = \{P_1, \ldots, P_N\}$ be a finite set and $\mathcal{P} \subset \{f \mid f : S \to \mathbb{F}_q\}$ such that $\mathcal{P}$ is a vector space over $\mathbb{F}_q$. Then we define $C = \mathbf{\Omega}(S, \mathcal{P})$ as the following subset of $(\mathbb{F}_q)^N$

$$\mathbf{\Omega}(S, \mathcal{P}) = \{(\sigma(P_1), \ldots, \sigma(P_N)) \,|\sigma \in \mathcal{P}\}.$$

It is obvious that $\mathbf{\Omega}(S, \mathcal{P})$ is a code in $(\mathbb{F}_q)^N$. We can obtain any nth-root code if we apply this construction to $S \subset R_n \cup \{0\}$ and $\mathcal{P} \subset \mathbb{F}_{q^m}[x]$.

To construct an AG code we need a projective, non singular, absolutely irreducible curve χ. We can take S as a subset of rational points of χ, so that $D = \sum_{i=1}^{N} P_i$ is a divisor on χ. To define $\mathcal{P}$ we have to choose another divisor G on χ such that $\mathrm{supp}(G) \cap D = \emptyset$ and then we consider

$$\mathcal{P} = \mathcal{L}(G) = \{f \in \mathbb{F}_q(\chi)^* | (f) \geq -G\} \cup \{0\}\,,$$

where $\mathbb{F}_q(\chi)$ is as usual the function field over $\mathbb{F}_q$. We thus obtain the AG code $C_{\mathcal{L}}(\chi, D, G) = \mathbf{\Omega}(S, \mathcal{P})$. It is possible to define the AG codes starting from rational differential forms and residues, but the previous construction is enough to describe any code as guaranteed by the following theorem.

Theorem 4.15 ([PSvW91])**.** *Any linear code is AG. However, there are linear codes that cannot be represented as* $C_{\mathcal{L}}(\chi, D, G)$ *with* $\deg(G) < N$.

Remark 4.16. An AG code as previously described is sometimes ([PSvW91]) called a **weakly** AG code.

The interest in the hypothesis $\deg(G) < N$ lies in the following theorem.

Theorem 4.17. *Let* $\mathtt{g}$ *be the genus of* χ *and* $\rho = \deg(G)$*. Let* $C = C_{\mathcal{L}}(\chi, D, G)$*. Then*

$$k \geq \rho + 1 - \mathtt{g}, \quad d \geq n - \rho, \ \textit{i.e.}$$
$$k + d \geq n - \mathtt{g} + 1\,.$$

Remark 4.18. Theorem 4.17 suggest to focus on low-genus curves for search for optimal codes.

From Proposition 2.8 it is clear that we can see any AG code (with $d \geq 2$) as an nth-root code, but it is not obvious how to do it. Actually, an explicit description for $\mathcal{L}(G)$ it is not known in general and that is one of the main problems while dealing with AG codes.

The most extensive research about AG codes has thus been carried out on **one-point** AG codes, i.e. codes such that $G = \{\rho P_\infty\}$, where P_∞ is the point at infinity of χ and D is the sum of all rational points of χ. In this case, we can view $\mathcal{L}(G) \subset \{f : (\mathbb{F}_q)^s \to \mathbb{F}_q\}$ and we can think of $\mathcal{L}(G)$ as $\mathcal{L} \subset \mathbb{F}_{q^s}[x]$ via some representation $\phi : (\mathbb{F}_{q^s}) \leftrightarrow (\mathbb{F}_q)^s$. Then, we take a linear basis of $\mathcal{L}(G)$,

$< \mathsf{g}_1, \ldots, \mathsf{g}_{N-k} > \subset \mathbb{F}_{q^s}[x]$, and we consider the following nth-root code

$$C = \Omega(q, q^s - 1, q^s, L, \mathcal{P}) \,,$$

where $L = \{P_1, \ldots, P_N\}$ is composed of all rational points of χ and $\mathcal{P} = \{\mathsf{g}_1, \ldots, \mathsf{g}_{N-k}\}$. The following example shows how these ideas can be applied to the most studied class of AG codes: the Hermitian codes.

Example 4.19. Let q be a power of a prime and χ be the Hermitian curve defined over $\mathbb{F}_{\mathsf{q}^2}$ by the affine equation $\chi : x^{\mathsf{q}+1} = y^{\mathsf{q}} + y$. This curve has genus $\mathsf{g} = \frac{\mathsf{q}(\mathsf{q}-1)}{2}$ and possesses $N = \mathsf{q}^3$ rational points, which we again call $P_1, \ldots, P_N$. Let ρ be a natural number such that $0 \leq \rho \leq n + 2\mathsf{g} - 2 = \mathsf{q}^3 + \mathsf{q}^2 - \mathsf{q} - 2$. The Hermitian code $C(\mathsf{q}, \rho)$ can be defined using the above construction, as follows. Let $D = \sum_{i=1}^{N} P_i$, $G = \rho P_\infty$ and $\mathcal{L}(G)$ be the corresponding vector subspace of rational functions on χ, then the Hermitian code $C(\mathsf{q}, \rho)$ (depending on q and ρ) is

$$C(\mathsf{q}, \rho) = \big\{ (f(P_1), \ldots, f(P_N)) \in (\mathbb{F}_{\mathsf{q}^2})^N \mid f \in \mathcal{L}(G) \big\} \,.$$

Set $\mathcal{L}(G)$ can be generated by a set of monomial functions

$$\mathcal{B} = \{ x^r y^s \mid \mathsf{q}r + (\mathsf{q}+1)s \leq \rho,\ 0 \leq r \leq \mathsf{q} - 1 \}$$

such that C has the following parity-check matrix

$$H = \begin{pmatrix} g_1(P_1) & \ldots & g_1(P_N) \\ \vdots & \ddots & \vdots \\ g_{N-k}(P_1) & \ldots & g_{N-k}(P_N) \end{pmatrix}$$

where $\{g_i\}_{1 \leq i \leq N-k}$ are $N - k$ monomials in $\mathcal{B}$.

With the nth-root construction we can see code C considered above as the nth-root code $\Omega(q, n, q^m, L, \mathcal{P})$, where the parameters are:

$$q = \mathsf{q}^2, n = \mathsf{q}^4 - 1, q^m = q^2, R_n = R_{\mathsf{q}^4 - 1} \cong (\mathbb{F}_{\mathsf{q}^2})^2 \setminus \{(0,0)\},$$

$L \cong \{(u, v) \in \chi \mid u, v \in \mathbb{F}_{\mathsf{q}^2}\}$, where the correspondence $\cong$ comes from the following (canonical) representation of finite fields

$$\phi : (\mathbb{F}_{\mathsf{q}^2})^2 \to \mathbb{F}_{\mathsf{q}^4} \qquad (u, v) \mapsto u + \beta v \,, \tag{11}$$

once β (a primitive element of $\mathbb{F}_{\mathsf{q}^4}$) is chosen. Then one can show ([Pel06])

$$L = \{ u + \beta v \mid (u, v) \in \chi \},$$

$$\mathcal{P} = \left\{ \left(\frac{\beta^{\mathsf{q}^2} z - \beta z}{\beta^{\mathsf{q}^2} - \beta} \right)^r, \left(\frac{z^{\mathsf{q}^2} - z}{\beta^{\mathsf{q}^2} - \beta} \right)^s \;\middle|\; \mathsf{q}r + (\mathsf{q}+1)s \leq \rho \right\} .$$

5 Complexity and computational considerations

The complexity of Gröbner basis computation has been the object of extensive studies. The worst case of their computation is double exponential ([Mor05]), but the generic behavior is much better. We recall some definitions and theorems, taken from [BFSY05], [BFS04], [BFS03] and [Bar01]. In the sequel, m,n and n are integers such that $\mathsf{m}, \mathsf{n} \geq 1$ and $\mathsf{k} = \mathsf{m} - \mathsf{n}$. Moreover, we will denote by R the polynomial ring $\mathbb{F}_2[y_1, \ldots, y_\mathsf{m}]$.

Definition 5.1. *Let $\{f_1, \ldots, f_\mathsf{m}\} \subset R$ be homogeneous polynomials. Polynomial sequence $(f_1, \ldots, f_\mathsf{m})$ is **regular** if for any $i = 1, \ldots, \mathsf{m}$, f_i is not a zero-divisor in the quotient ring $R/\langle\{f_1, \ldots, f_{i-1}\}\rangle$. In other words,*

$$g \in R,\, gf_i \in \langle\{f_1, \ldots, f_{i-1}\}\rangle \implies g \in \langle\{f_1, \ldots, f_{i-1}\}\rangle\,.$$

Regular systems do not exist when the number of polynomials, m, is larger than the number of variables, n. To overcome this difficulty, M. Bardet extends this notion in her thesis, as follows. Let from now on $(f_1, \ldots, f_\mathsf{m})$ denote a polynomial sequence in R such that ideal $\langle\{f_1, \ldots, f_\mathsf{m}\}\rangle$ is zero-dimensional and all f_i's are homogeneous (this implies $\mathsf{m} \geq \mathsf{n}$).

Definition 5.2. *Polynomial sequence $(f_1, \ldots, f_\mathsf{m})$ is **$\mathbf{d}$-regular** if, for any $i = 1, \ldots, \mathsf{m}$, we have*

$$g \in R,\, \deg(g) < d - d_i,\, gf_i \in \langle\{f_1, \ldots, f_{i-1}\}\rangle \implies g \in \langle\{f_1, \ldots, f_{i-1}\}\rangle\,.$$

Definition 5.3. *We define the **degree of regularity** D_{reg} of ideal $\langle\{f_1, \ldots, f_\mathsf{m}\}\rangle$ as*

$$D_{\mathrm{reg}} = \min\left\{\mathsf{d} \geq 0 \mid \dim_{\mathbb{F}_2}(\{f \in I, \deg(f) = \mathsf{d}\}) = \binom{\mathsf{n}+\mathsf{d}}{\mathsf{d}}\right\}.$$

Any D_{reg}-regular system is called **semi-regular**.

Theorem 5.4. *The degree of regularity of a semi-regular system of $\mathsf{m} = \mathsf{n} + \mathsf{k}$ homogeneous polynomials of degree $\mathsf{d}_1 = \deg(f_1), \ldots, \mathsf{d}_{\mathsf{n}+\mathsf{k}} = \deg(f_{\mathsf{n}+\mathsf{k}})$ in n variables behaves asymptotically (with k constant and $\mathsf{n} \to \infty$) as*

$$D_{\mathrm{reg}} = \sum_{i=1}^{\mathsf{m}} \frac{\mathsf{d}_i - 1}{2} - \alpha_\mathsf{k}\sqrt{\sum_{i=1}^{\mathsf{m}} \frac{\mathsf{d}_i^2 - 1}{6}} + O(1)\,,$$

where α_k is the largest zero of the k-th Hermite polynomial.

There exist asymptotic estimations for the largest zero α_k of the k-th Hermite polynomial ([ADGR04]), i.e.

$$\alpha_\mathsf{k} \overset{\mathsf{k}\to\infty}{\longrightarrow} \sqrt{2}\sqrt{\mathsf{k}}. \tag{12}$$

We now explain why the regularity degree is so important to estimate the complexity of Gröbner basis computation. The best-known algorithm to calculate Gröbner bases, Faugere's F5, is essentially based on the determination of an echelon form for Macaulay matrices. A Macaulay matrix of degree D is constructed starting from a set of homogeneous polynomials $\{f_1, \ldots, f_z\}$, as follows. We multiply any f_i for some monomials m_j such that $\deg(f_i m_j) = D$ The choice of these monomials depends on the algorithm optimizations. Let $\{F_h\}_{1 \leq h \leq z'}$ be the polynomials so obtained. We then construct a matrix with z' rows and with a number of columns equal to the number of all monomials of degree D. Any row of the matrix corresponds to an F_h and its entries are nothing else than the coefficients of the corresponding monomial in F_h. The cost of F5 is dominated by the cost of linear algebra on the biggest Macaulay matrix needed, which is the matrix corresponding to the degree of regularity, as shown in Bardet's thesis.

We would like to estimate the degree of regularity of $J_w(C)$ (or better, of a sequence equivalent to it) in the binary case, where w is understood from now on to be $w \geq 2$. For simplicity we consider only the zerofree case (but the non-zerofree can be shown to behave identically) and we restrict to the maximal case, since it is the worst for us (the degrees of the input polynomials are higher). We denote by $\mathbb{F}_{2^m}[Z]$ the polynomial ring $\mathbb{F}_{2^m}[z_1, \ldots, z_w]$.

Let $C = \Omega(2, n, 2^m, L, \mathcal{P})$ be a binary zero-free maximal nth-root code. The polynomial basis of ideal J_w given in Remark 2.12 is not homogenous, so we introduce a new variable ζ and homogenize. The homogeneous basis so obtained then gives rise to the following ideal $\mathfrak{J}_w = \mathfrak{J}_w(C)$

$$\mathfrak{J}_w = \langle \Big\{ \sum_{k=1}^{w} \bar{g}_t(z_k, \zeta) \Big\}_{1 \leq t \leq r},\ \{p_{i,j}(z_i, z_j)\}_{1 \leq i \neq j \leq w},\ \{z_j^n - \zeta^n\}_{1 \leq j \leq w} \rangle \qquad (13)$$

where $\bar{g}_t(z, \zeta)$ is the homogenized polymonial obtained by $g_t(z)$, for any $g_t \in \mathcal{P}$.

In order to apply Theorem 5.4 to system $\mathfrak{J}_w$, we note the following:

- we may assume $\mathfrak{J}_w$ to be semi-regular for w large enough, since a generic sequence of polynomials is conjectured to be semi-regular with a large number of variables (many computer experiments with random sequences and our own experiments show the same), being many special case already formally proved;
- however, the number of variables is $\mathsf{n} = w + 1$ and the number of polynomials is $\mathsf{m} = r + \binom{w}{2} + w$, so that $\mathsf{k} = \mathsf{m} - \mathsf{n} = \left(r + \binom{w}{2} + w\right) - (w + 1) = r + \binom{w}{2} - 1$.

Since k is not constant w.r.t. n, we cannot apply Theorem 5.4. No formulae of such type are known for this case and so a direct application of Bardet's theory is not feasible. In the next subsection we introduce a modified ideal that gives anyway our desired result and to which Theorem 5.4 can be safely applied. This application will be done in Subsection 5.2.

5.1 Spurious solutions in the binary case

In this subsection we accelerate the computation of Gröbner basis $\mathcal{G}$ of ideal $J_w(C)$ by removing polynomials $p_{i,j}(z_i, z_j)$, which guarantee $z_i \neq z_j$ for any $i \neq j$. This gives rise to spurious solutions, that may be counted with elementary combinatorial arguments. Although it is possible to treat the general case, the involved arguments soon become long and cumbersome. Here we restrict to the computation of the number of minimum weight codewords (and the distance), when the code is binary.

Throughout this subsection w, N and m are three integers such that $1 \leq w \leq N$ and $m \geq 1$. We also denote by $\mathbb{F}_{2^m}[Z]$ the polynomial ring $\mathbb{F}_{2^m}[z_1, \ldots, z_w]$, and by $\overline{\mathbb{F}}$ the algebraic closure of $\mathbb{F}_2$.

Definition 5.5. *Let $C = \Omega(2, n, 2^m, L, \mathcal{P})$ be a binary nth-root code, with $|L| = N$. We denote by $I_w = I_w(C)$ the following ideal in $\mathbb{F}_{2^m}[Z]$*

$$I_w = \left\langle \left\{ \sum_{k=1}^{w} g_t(z_k) \right\}_{1 \leq t \leq r}, \quad \left\{ \frac{z_j^n - 1}{\prod_{l \in \bar{L}} (z_j - l)} \right\}_{1 \leq j \leq w} \right\rangle \tag{14}$$

For the remainder of this subsection, C is understood.

Remark 5.6. It is obvious that $J_w(C) = \langle I_w(C), \{p_{i,j}(z_i, z_j)\}_{1 \leq i \neq j \leq w} \rangle$, so that $V \in \mathcal{V}(J_w(C))$ if and only if $V \in \mathcal{V}(I_w(C))$ and all components of V are distinct. Furthermore, we can easily extend Definition 2.10 ($J_w(C)$) to the case $w = 1$ by simply setting $J_1(C) = I_1(C)$ (and hence $\mathcal{V}(J_1(C)) = \mathcal{V}(I_1(C))$).

Definition 5.7. *Let $\{\mathcal{J}_{\mathsf{w}}\}_{\mathsf{w} \geq 1}$ be the following ideal sequence*

$$\mathcal{J}_{\mathsf{w}} \subset \mathbb{F}_{2^m}[z_1, \ldots, z_{\mathsf{w}}], \qquad \mathcal{J}_{\mathsf{w}} = \langle \{l_i^{\mathsf{w}}\}_{i=1,\ldots,r}, \quad \{H(z_j)\}_{j=1,\ldots,\mathsf{w}} \rangle,$$

where $l_i^{\mathsf{w}} \in \mathbb{F}_{2^m}[z_1, \ldots, z_{\mathsf{w}}]$ for any $\mathsf{w} \geq 1$ and any $1 \leq i \leq r$, and $H \in \mathbb{F}_{2^m}[z]$.

*We say that $\{\mathcal{J}_{\mathsf{w}}\}$ is **self-related** if*

(1) polynomial l_i^{w} is symmetric for any $i = 1, \ldots, r$ and any $\mathsf{w} \geq 2$;

(2) $l_i^{\mathsf{w}}(\bar{z}_1, \ldots, \bar{z}_{\mathsf{w}-2}, \mathbf{z}, \mathbf{z}) = 0 \Leftrightarrow l_i^{\mathsf{w}-2}(\bar{z}_1, \ldots, \bar{z}_{\mathsf{w}-2}) = 0$ for any $i = 1, \ldots, r$, any $\mathsf{w} \geq 3$ and any $\bar{z}_1, \ldots, \bar{z}_{\mathsf{w}-2}, \mathbf{z} \in \overline{\mathbb{F}}$;

(3) $V \in \mathcal{V}(\mathcal{J}_2) \Rightarrow V = (\bar{z}, \bar{z})$ for some $\bar{z} \in \overline{\mathbb{F}}$;

(4) $\mathcal{V}(\mathcal{J}_1) = \emptyset$.

From now on $\{\mathcal{J}_w\}$ is understood to be a generic ideal sequence $\{\mathcal{J}_w\}_{w \geq 1}$ such that $\mathcal{J}_w \subset \mathbb{F}_{2^m}[Z]$ for any $w \geq 1$.

Fact 5.8. *Ideal sequence $\{I_w(C)\}$ (Definition 5.5) is self-related if $d(C) \geq 3$.*

Proof. We show all conditions (1)–(4) in Definition 5.7.

(1) Polynomial $l_i^w = \sum_{k=1}^{w} g_i(z_k)$ is obviously symmetric for $i = 1, \ldots, r$.

(2) For any $\mathbf{z} \in \overline{\mathbb{F}}$ and any $i = 1, \ldots, r$, $g_i(\mathbf{z}) + g_i(\mathbf{z}) = 0$, so that

$$l_i^w(\bar{z}_1, \ldots, \bar{z}_{w-2}, \mathbf{z}, \mathbf{z}) = \sum_{k=1}^{w-2} g_i(\bar{z}_k) + g_i(\mathbf{z}) + g_i(\mathbf{z}) = \sum_{k=1}^{w-2} g_i(\bar{z}_k) = l_i^{w-2}(\bar{z}_1, \ldots, \bar{z}_{w-2}) .$$

(3) If $V \in \mathcal{V}(I_2)$ is of type $(\bar{z}, \dot{z})$, with $\bar{z} \neq \dot{z}$, then $V \in \mathcal{V}(J_2)$ (Remark 5.6) and so there is in C at least a codeword of weight 2, which is not possible since $d \geq 3$. Thus, $\bar{z} = \dot{z}$.

(4) If $V \in \mathcal{V}(I_1)$, then $V \in \mathcal{V}(J_1)$ (Remark 5.6), and so there is in C at least a codeword of weight 1, which is impossible. □

Definition 5.9. *Let $V = (\bar{z}_1, \ldots, \bar{z}_w) \in (\overline{\mathbb{F}})^w$, with $w \geq 2$. We say that*

*(1) V is **weakly double-coordinate** (**wdc**) if there exist $i, j = 1, \ldots, w$, $i \neq j$, such that $\bar{z}_i = \bar{z}_j$;*

*(2) V is **strongly double-coordinate** (**sdc**) if w is even and for any i, $1 \leq i \leq w$, there is j such that $\bar{z}_i = \bar{z}_j$.*

We can obviously extend the definition of wdc (and sdc) vectors to w-tuples of a generic Cartesian product.

Definition 5.10. *For any w, if $\mathcal{V}(\mathcal{J}_w) = \emptyset$ we say that $\mathcal{J}_w$ is a wdc ideal and a sdc ideal.*

For any $w \geq 2$, we say that $\mathcal{J}_w$ is a wdc ideal if all its solutions are wdc. If $w \geq 2$ is even, we say that $\mathcal{J}_w$ is a sdc ideal if all its solutions are sdc.

Remark 5.11. If a vector (or an ideal) is sdc, then it is wdc.

If $w = 2$ the notions of wdc and sdc are coincident.

Lemma 5.12. *Let $\{\mathcal{J}_w\}$ be a self-related ideal sequence. Suppose that $\mathcal{J}_{w'}$ is wdc for all $1 \leq w' \leq w$, then ideal $\mathcal{J}_{w'}$ is sdc for any $w' \leq w$.*

Proof. We first show $\mathcal{J}_w$ is sdc if w is even.
We prove this by induction on w.
If $w = 2$ then $\mathcal{J}_2$ is both wdc and sdc thanks to Remark 5.11.
We now suppose that the assertion holds for $w - 2$ and we prove it for w.
For any $w' \leq w - 2$, $\mathcal{J}_{w'}$ is wdc, so by induction hypothesis $\mathcal{J}_{w-2}$ is sdc. Let $V = (\bar{z}_1, \ldots, \bar{z}_w) \in \mathcal{V}(\mathcal{J}_w)$ be any solution. As $\mathcal{J}_w$ is wdc, V has two components with the same value, for example $\bar{z}_w = \bar{z}_{w-1}$. We truncate V in the last two components, obtaining $\tilde{V} = (\bar{z}_1, \ldots, \bar{z}_{w-2}) \in \mathcal{V}(\mathcal{J}_{w-2})$ (thanks to condition 2 in Definition 5.7). But $\mathcal{J}_{w-2}$ is sdc, so $\tilde{V}$ is sdc and hence V is sdc. Since V is arbitrary, also $\mathcal{J}_w$ is sdc.

We now show $\mathcal{J}_w$ is sdc if w is odd.
We prove this by induction on w.
If $w = 1$ then $\mathcal{J}_1$ is both wdc and sdc thanks to (4) in Definition 5.7.
We now suppose that the assertion holds for $w - 2$ and we prove it for w.

For any $w' \leq w-2$, $\mathcal{J}_{w'}$ is wdc, so by induction hypothesis $\mathcal{J}_{w-2}$ is sdc, i.e. it has no solution. Let V and $\tilde{V}$ be as in the even case. Again, $\tilde{V}$ should lie in $\mathcal{V}(\mathcal{J}_{w-2})$ (thanks to condition 2 in Definition 5.7), which is empty and so V does not exist. Since V is arbitrary, also $\mathcal{J}_w$ is sdc.

The general case readily follows from the two previous ones. □

Lemma 5.13. *Ideals $I_w(C)$ are wdc for all $w \leq d-1$.*

Proof. Solutions of system $I_w(C)$ for any $w \leq d-1$ cannot correspond to codewords (since no weight-w codeword exists) and hence they are spurious, which means they have two coincident components, i.e. they are wdc. □

Definition 5.14. *Let $H \in \mathbb{F}_{2^m}[z]$. For any w, we denote by $\mathcal{A}_w(H)$ the set of all sdc vectors in $(\mathcal{V}(H))^w \cap (\mathbb{F}_{2^m})^w$.*

Theorem 5.15. *Let $d \geq 3$. Then, ideal $I_w(C)$ is sdc for any $1 \leq w \leq d-1$. Moreover:*

- *if d is odd, $\mathcal{V}(I_d(C)) = \mathcal{V}(J_d(C))$,*
- *if d is even, $\mathcal{V}(J_d(C)) = \mathcal{V}(I_d(C)) \sqcup \mathcal{A}_d(H')$, where $H' = \frac{z^n-1}{\prod_{l \in \bar{L}}(z-l)}$.*

Proof. Since $w \leq d-1$, Lemma 5.13 and Lemma 5.12 imply that I_w is sdc.

Let us suppose d odd. If V is in $\mathcal{V}(I_d(C)) \backslash \mathcal{V}(J_d(C))$, then it is a spurious solution. Let $\tilde{V}$ as in the proof of Lemma 5.12. We will have $\tilde{V} \in \mathcal{V}(J_{d-2}(C))$. But $d-2 < d$ and $d-2$ is odd, so that by the first part of our proof $\mathcal{V}(J_{d-2}(C)) = \emptyset$. Hence, such V cannot exist and $\mathcal{V}(I_d(C)) \setminus \mathcal{V}(J_d(C)) = \emptyset$.

Let us suppose d even. Any spurious solution is wdc. It is enough to show that any vector in $\mathcal{A}_w(H)$ is in $\mathcal{V}(I_d(C))$, since by construction of $I_d(C)$ no other spurious solution can exist.
Let $V \in \mathcal{A}_w(H)$, $V = (v_1, \ldots, v_d)$. Its component obviously satisfy $H(v_i) = 0$ (for any i). On the other hand, we can group components $\{v_i\}$ according to their values, so that $\{1, \ldots, d\} = \sqcup_{\iota=1}^{\iota'} S_\iota$, where $v_i = v_j$ if and only if $i, j \in V_\iota$ for one and only one ι. We then have, for any $1 \leq i \leq r$,

$$\sum_{k=1}^{w} g_i(v_k) = \sum_{\iota=1}^{\iota'} \left(\sum_{k \in S_\iota} g_i(v_k) \right) = \sum_{\iota=1}^{\iota'} 0 = 0\,.$$

□

To count the number of spurious solutions we provide the following general recursive formula.

Fact 5.16. *Let l be an even integer $l \geq 2$ and λ be an integer $\lambda \geq 1$. Let $\mathcal{T} = \{\zeta_1, \ldots, \zeta_\lambda\}$ be any set with $|\mathcal{T}| = \lambda$. Let $\mathcal{T}^l$ be the standard Cartesian product. Let $a(l, \lambda)$ be the number of sdc l-tuples in $\mathcal{T}^l$. For any integers $\lambda', l' \geq 1$,*

define $a(0,\lambda')=1$ *and* $a(l',1)=1$. *Then*

$$a(l,\lambda)=\sum_{s=0}^{l/2}\binom{l}{2s}a(l-2s,\lambda-1)\,. \tag{15}$$

Proof. Let $v=(v_1,\ldots,v_l)\in\mathcal{T}^l$. Element ζ_λ can appear in v either 2 or 4 or $\ldots$ l times. If ζ_λ is in exactly 2 components of v, say v_i and v_j, the $(l-2)$-tuple $\tilde{v}$ obtained by puncturing v in positions i and j is a sdc $(l-2)$-tuple in $\{\zeta_1,\ldots,\zeta_{\lambda-1}\}^{l-2}$. Moreover, i and j can be any two positions. Thus, the number of sdc l-tuples v having exactly 2 components equal to an assigned value (e.g., to ζ_λ) is $\binom{l}{2}a(l-2,\lambda-1)$.
Analogously, the number of sdc l-tuples having exactly 4 components equal to an assigned value is $\binom{l}{4}a(l-4,\lambda-1)$. By summing all these values, we obtain our claimed expression. □

Since

$$\mathcal{V}\left(\frac{z^n-1}{\prod_{l\in\bar{L}}(z-l)}\right)\subset\mathbb{F}_{2^m}\ ,$$

by Theorem 5.15 and Proposition 2.13, we have our final result for this subsection.

Corollary 5.17. *Let* $C=\Omega(2,n,2^m,L,\mathcal{P})$ *be a binary zero-free nth-root code. Then* A_d *is:*

$$A_d=\frac{|\mathcal{V}(I_d)(C)|-a(d,N)}{d!}\qquad (d\text{ even}),$$

$$A_d=\frac{|\mathcal{V}(I_d)(C)|}{d!}\qquad (d\text{ odd}).$$

Example 5.18. Let $C=\Omega(2,255,2^8,L,\mathcal{P})$ be the binary nth-root code such that $L=\mathbb{F}_{256}\setminus\{0\}$ and $\mathcal{P}=\{x,x^2,x^3,x^4,x^5,x^6\}\subset\mathbb{F}_2[x]$. We have $n=N=255$ and C is nothing else that a BCH code with designed distance 7. In particular, it cannot have words of weight 5. By computing a Gröbner basis of $I_5(C)$ and $J_5(C)$, we obtain that $|I_5(C)|=|J_5(C)|=0$, so that $A_5(C)=0$, as expected, but the computations in the $I_5(C)$ case takes less than 4 seconds, while the computations of the $J_5(C)$ case takes 28 seconds.

Remark 5.19. In a personal communication, F. Caruso claims the following explicit formula to compute $a(l,t)$

$$a(l,\lambda)=\frac{1}{2^{\lambda-1}}\sum_{j=0}^{\lceil\lambda/2\rceil-1}\binom{\lambda}{j}(\lambda-2j)^l\,.$$

5.2 Regularity degree with spurious solutions

We now apply Bardet's theory to the situation studied in the previous subsection.

Let C be binary maximal zerofree. Let $\mathfrak{I}_w = \mathfrak{I}_w(C)$ be the ideal obtained by homogenizing the input basis of $I_w(C)$, as follows:

$$\mathfrak{I}_w = \langle \Big\{ \sum_{k=1}^{w} \bar{g}_t(z_k, \zeta) \Big\}_{1 \le t \le r},\ \{z_j^n - \zeta^n\}_{1 \le j \le w} \rangle \tag{16}$$

where $\bar{g}_t(z, \zeta)$ is the homogenized polynomial obtained by $g_t(z)$, for any $g_t \in \mathcal{P}$. We have m polynomials and n variables, with

$$\mathsf{n} = w + 1, \quad \mathsf{m} = w + r, \quad \mathsf{k} = \mathsf{m} - \mathsf{n} = r - 1\,.$$

We can then apply Theorem 5.4 to $\mathfrak{I}_w$ by considering r fixed (but large enough to apply (12)) and $\mathcal{P}$ generic in $\mathbb{F}_{2^m}[x]$, with m growing (so that $n = 2^m - 1$ can grow and $w \le n$). We then have

$$D_{\text{reg}} = D_{\text{reg}}(\mathfrak{I}_w) = \sum_{i=1}^{r} \frac{d_i - 1}{2} + \sum_{i=1}^{w} \frac{n-1}{2} - \alpha_{r-1} \sqrt{\sum_{i=1}^{r} \frac{d_i^2 - 1}{6} + \sum_{i=1}^{w} \frac{n^2 - 1}{6}}$$

where $d_i = \deg(g_i)$ and $n = \deg(z_j^n - \zeta^n)$.

We now estimate D_{reg} when w goes to infinity:

$$\lim_{w \longrightarrow \infty} D_{\text{reg}} = wn \lim_{w \longrightarrow \infty} \frac{D_{\text{reg}}}{wn}\,.$$

We know $\mathrm{d}_i \le n$, so that $\sum_{i=1}^{r} \frac{d_i - 1}{2} \le rn$ and hence (r is constant)

$$\lim_{w \longrightarrow \infty} \frac{1}{wn} \left(\sum_{i=1}^{r} \frac{d_i - 1}{2} \right) \le \lim_{w \longrightarrow \infty} \frac{1}{wn} rn = 0\,. \tag{17}$$

Similarly, inside the square root,

$$\lim_{w \longrightarrow \infty} \frac{1}{w^2 n^2} \left(\alpha_{r-1}^2 \sum_{i=1}^{r} \frac{d_i^2 - 1}{6} \right) \le \lim_{w \longrightarrow \infty} \frac{2r}{w^2 n^2} r n^2 = 0\,. \tag{18}$$

The remaining terms give

$$D_{\text{reg}} \sim \frac{wn}{2} - \alpha_{r-1} \sqrt{\frac{wn^2}{6}}$$

By applying (12) for r large enough, we finally obtain

$$D_{\text{reg}} \sim \frac{wn}{2} - n \sqrt{\frac{w(r-1)}{3}} \sim \frac{wn}{2}\,. \tag{19}$$

6 Conclusions and further research

Linear codes are traditionally specified starting from a parity-check matrix H. In particular, cyclic codes are such that the entries of H consist of the evaluation of univariate monomials on all the n-th roots of unity. Our approach in this paper is to specify "any" linear code (with $d \geq 2$) as a code such that the entries of H consist of the evaluation of generic (univariate) polynomials on all the n-th roots of unity. In this sense, we say that linear codes "are" a generalization of cyclic codes.

This point of view allows to extend to linear codes some computational algebra techniques and some argument, that have been previously applied to cyclic codes. This translates in new tools, but also in new challenges. To be more precise, we can identify two main tools, both based on Groebner basis computations, i.e. :

- algorithms to compute the weight distribution (and the distance),
- a new decoding algorithm for a (potentially very large) sub-class, via the general error locator polynomial.

Let us consider the first tool. The problem of determining the weight distribution of a code is an NP-hard problem ([BD92], [Bar98]). We cannot expect from our algorithm any computational improvement on known algorithms for (generic) linear codes. However, the notion of a "generic linear code" is not widely accepted, except in the sense that the code does not belong to any known family (but recall that *any* linear code can be seen both as a weakly AG code and as an affine-variety code, making the notion of *known family* rather questionable). If instead you view your code as an nth-root code, some algebraic properties may become apparent. Indeed, as it is clear from our examples, a code can be seen an nth-root code in many different ways, some of them leading to interesting properties. This is even more clear if you look at the second tool. The problem of decoding linear codes is NP-hard ([Bar98], [BKvT99]), but if a linear code admits a sparse general error locator polynomial (or such a polynomial with a sparse representation), then it can be decoded very fast. We have provided an explicit example when the locator polynomial is very small, given a certain nth-root presentation, and long when given another. Yet, the code in consideration does not belong to any known family. In other words, the question "what can we do with a generic linear code?" becomes now "what is a generic linear code?". If we define a generic linear code as a code such that our tools can be applied efficiently, then it becomes worthwhile to try showing that "most" codes satisfy this definition (which we believe to be true). We have thus identified a research problem:

given a linear code, either find an nth-root presentation such that our tools can be efficiently applied or show that such presentation does not exist.

Acknowledgments

The first author would like to thank the second author (her supervisor). Part of this work has been presented at "Workshop D1: Gröbner Bases in Cryptography, Coding Theory, and Algebraic Combinatorics", Linz, 2006, and at "Workshop on Coding and Cryptography", UCC, Cork, 2006.

The authors would like to thank the following people for their comments and suggestions: J. Abbot, M. Bardet, F. Caruso, F. Dalla Volta, J. C. Faugere, P. Fitzpatrick, T. Mora, E. Orsini, M. Pellegrini, L. Perret, I. Simonetti, C. Traverso.

We have run our computer simulations using the software package Singular (`http://www.singular.uni-kl.de`) at the computational centre MEDICIS (`http://medicis.polytechnique.fr`).

This work has been partially supported by the STMicroelectronics contract "Complexity issues in algebraic Coding Theory and Cryptography".

References

[ADGR04] I. Area, D. K. Dimitrov, E. Godoy, and A. Ronveaux, *Zeros of Gegenbauer and Hermite polynomials and connection coefficients*, Math. Comp. **73** (2004), no. 248, 1937–1951.

[Bar98] A. Barg, *Complexity issues in coding theory*, Handbook of coding theory, Vol. I, II, North-Holland, Amsterdam, 1998, pp. 649–754.

[Bar01] Magali Bardet, *An investigation on overdetermined algebraic systems and applications to error-correcting codes and to cryptography*, Ph.D. thesis, University of Paris 6, Paris, France, 2001.

[BCRT93] A. M. Bigatti, P. Conti, L. Robbiano, and C. Traverso, *A "divide and conquer" algorithm for Hilbert-Poincaré series, multiplicity and dimension of monomial ideals*, Applied algebra, algebraic algorithms and error-correcting codes (San Juan, PR, 1993), Lecture Notes in Comput. Sci., vol. 673, Springer, Berlin, 1993, pp. 76–88.

[BD92] A. M. Barg and I. I. Dumer, *On computing the weight spectrum of cyclic codes*, IEEE Trans. Inform. Theory **38** (1992), no. 4, 1382–1386.

[BFS03] M. Bardet, J. C. Faugère, and B. Salvy, *Complexity of Groebner basis computation for semi-regular overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$*, Inria Research Report RR-5049, INRIA, France, 2003.

[BFS04] M. Bardet, J. C. Faugere, and B. Salvy, *On the complexity of Groebner basis computation of semi-regular overdetermined algebraic equations*, Tech. report, Talk at ICPSS 2004, 2004.

[BFSY05] M. Bardet, J. C. Faugere, B. Salvy, and B. Y. Yang, *Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems*, Tech. report, Talk at MEGA 2005, 2005.

[BKvT99] A. Barg, E. Krouk, and H. C. A. van Tilborg, *On the complexity of minimum distance decoding of long linear codes*, IEEE Trans. Inform. Theory **45** (1999), no. 5, 1392–1405.

[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.

[BS06] E. Betti and M. Sala, *A new bound for the minimum distance of a cyclic code from its defining set*, IEEE Trans. Inform. Theory **52** (2006), no. 8, 3700–3706.

[CM02] M. Caboara and T. Mora, *The Chen-Reed-Helleseth-Truong decoding algorithm and the Gianni-Kalkbrenner Groebner shape theorem*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 3, 209–232.

[Fit95] P. Fitzpatrick, *On the key equation*, IEEE Trans. Inform. Theory **41** (1995), no. 5, 1290–1302.

[FL98] J. Fitzgerald and R. F. Lax, *Decoding affine variety codes using Groebner bases*, Des. Codes Cryptogr. **13** (1998), no. 2, 147–158.

[Gia89] P. Gianni, *Properties of Groebner bases under specializations*, EUROCAL '87 (Leipzig, 1987), Lecture Notes in Comput. Sci., vol. 378, Springer, Berlin, 1989, pp. 293–297.

[GM89] P. Gianni and T. Mora, *Algebraic solution of systems of polynomial equations using Groebner bases*, Applied algebra, algebraic algorithms and error-correcting codes (Menorca, 1987), Lecture Notes in Comput. Sci., vol. 356, Springer, Berlin, 1989, pp. 247–257.

[HT74] C. R. P. Hartmann and K. K. Tzeng, *Decoding beyond the BCH bound using multiple sets of syndrome sequences*, IEEE Trans. Inform. Theory **20** (1974), 292–295.

[Kal89] M. Kalkbrener, *Solving systems of algebraic equations by using Groebner bases*, EUROCAL '87 (Leipzig, 1987), Lecture Notes in Comput. Sci., vol. 378, Springer, Berlin, 1989, pp. 282–292.

[KM00] Y. Katayama and S. Morioka, *One-shot reed-solomon decoding for high-performance dependable systems*, IEEE DSN **00** (2000), 390.

[Mor05] T. Mora, *Solving polynomial equation systems. II*, Encyclopedia of Mathematics and its Applications, vol. 99, Cambridge University Press, Cambridge, 2005, Macaulay's paradigm and Gröbner technology.

[MOS06] T. Mora, E. Orsini, and M. Sala, *General error locator polynomials for binary cyclic codes with* $t <= 2$ *and* $n < 63$, BCRI preprint, `www.bcri.ucc.ie` 43, University College Cork, Boole Centre BCRI, UCC Cork, Ireland, 2006.

[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.

[MS03] T. Mora and M. Sala, *On the Groebner bases of some symmetric systems and their application to coding theory*, J. Symbolic Comput. **35** (2003), no. 2, 177–194.

[OS05] E. Orsini and M. Sala, *Correcting errors and erasures via the syndrome variety*, J. Pure Appl. Algebra **200** (2005), no. 1-2, 191–226.

[OS06] ———, *General error locator polynomials for binary cyclic codes with* $t \leq 2$ *and* $n < 63$, IEEE Trans. Inform. Theory (2006), Accepted for publication.

[Pel06] M. Pellegrini, *On the weight distribution of some goppa ag codes*, Ph.D. thesis, University of Pisa, 2006, Work in progress.

[PHB98] V. S. Pless, W. C. Huffman, and R. A. Brualdi (eds.), *Handbook of coding theory. Vol. I, II*, North-Holland, Amsterdam, 1998.

[PSvW91] R. Pellikaan, B. Z. Shen, and G. J. M. van Wee, *Which linear codes are algebraic-geometric?*, IEEE Trans. Inform. Theory **37** (1991), no. 3, part 1, 583–602.

[PW72] W. W. Peterson and E. J. Weldon, Jr., *Error-correcting codes*, second ed., The M.I.T. Press, Cambridge, Mass.-London, 1972.

[Roo83] C. Roos, *A new lower bound for the minimum distance of a cyclic code*, IEEE Trans. Inform. Theory **29** (1983), no. 3, 330–332.

[Sal02] M. Sala, *Groebner bases and distance of cyclic codes*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 2, 137–162.

[Sal06] ———, *Groebner basis techniques to compute weight distributions of shortened cyclic codes*, Journal of Algebra and Its Applications (2006), Accepted for publication.

[Sei74] A. Seidenberg, *Constructions in algebra*, Trans. Amer. Math. Soc. **197** (1974), 273–313.

[ST00] M. Sala and A. Tamponi, *A linear programming estimate of the weight distribution of* BCH$(255, k)$, IEEE Trans. Inform. Theory **46** (2000), no. 6, 2235–2237.

7 Appendix A

Output with *redSB*

$$
\begin{aligned}
&J1[194] = z_1 + w_2w_1x_2^3x_1^6 + x_2^5x_1^6 + w_2^3x_2x_1^7 + w_1^3x_2x_1^7 + w_2^4w_1^3x_2^3 + w_2^3w_1^4x_2^3 + \\
&w_2^5w_1x_2^4 + w_2^4w_1^2x_2^4 + w_2^2w_1^4x_2^4 + w_2w_1^5x_2^4 + w_2^3w_1^2x_2^5 + w_2^2w_1^3x_2^5 + w_2^4x_2^6 + \\
&w_2^3w_1x_2^6 + w_2^2w_1^2x_2^6 + w_2w_1^3x_2^6 + w_1^4x_2^6 + w_2^2w_1x_2^7 + w_2w_1^2x_2^7 + w_2^5w_1^2x_2^2x_1 + \\
&w_2^2w_1^5x_2^2x_1 + w_2^4w_1^2x_2^3x_1 + w_2^3w_1^3x_2^3x_1 + w_2^2w_1^4x_2^3x_1 + w_2^4w_1x_2^4x_1 + \\
&w_2w_1^4x_2^4x_1 + w_2^4x_2^5x_1 + w_2^2w_1^2x_2^5x_1 + w_1^4x_2^5x_1 + w_2^2w_1x_2^6x_1 + w_2w_1^2x_2^6x_1 + \\
&w_2^4w_1^3x_2x_1^2 + w_2^3w_1^4x_2x_1^2 + w_2^5w_1x_2^2x_1^2 + w_2w_1^5x_2^2x_1^2 + w_2^5x_2^3x_1^2 + \\
&w_2^4w_1x_2^3x_1^2 + w_2^3w_1^2x_2^3x_1^2 + w_2^2w_1^3x_2^3x_1^2 + w_2w_1^4x_2^3x_1^2 + w_1^5x_2^3x_1^2 + \\
&w_2^3w_1x_2^4x_1^2 + w_2w_1^3x_2^4x_1^2 + w_2^2w_1x_2^5x_1^2 + w_2w_1^2x_2^5x_1^2 + w_2^2x_2^6x_1^2 + \\
&w_2w_1x_2^6x_1^2 + w_1^2x_2^6x_1^2 + w_2x_2^7x_1^2 + w_1x_2^7x_1^2 + w_1^5w_1x_2x_1^3 + w_2^4w_1^2x_2x_1^3 + \\
&w_2^2w_1^4x_2x_1^3 + w_2w_1^5x_2x_1^3 + w_2^5x_2^2x_1^3 + w_1^5x_2^2x_1^3 + w_2^4x_2^3x_1^3 + w_1^4x_2^3x_1^3 + \\
&w_2w_1x_2^5x_1^3 + w_2x_2^6x_1^3 + w_1x_2^6x_1^3 + x_2^7x_1^3 + w_2^5w_1x_1^4 + w_2^4w_1^2x_1^4 + w_2^3w_1^3x_1^4 + \\
&w_2^2w_1^4x_1^4 + w_2w_1^5x_1^4 + w_2^5x_2x_1^4 + w_2^4w_1x_2x_1^4 + w_2w_1^4x_2x_1^4 + w_1^5x_2x_1^4 + \\
&w_2^4x_2^2x_1^4 + w_2^3w_1x_2^2x_1^4 + w_2w_1^3x_2^2x_1^4 + w_1^4x_2^2x_1^4 + w_2^3x_2^3x_1^4 + w_2^2w_1x_2^3x_1^4 + \\
&w_2w_1^2x_2^3x_1^4 + w_1^3x_2^3x_1^4 + w_2^2x_2^4x_1^4 + w_1^2x_2^4x_1^4 + w_2x_2^5x_1^4 + w_1x_2^5x_1^4 + \\
&w_2^2w_1^2x_2x_1^5 + w_2^2w_1x_2^2x_1^5 + w_2w_1^2x_2^2x_1^5 + w_2^2x_2^3x_1^5 + w_2w_1x_2^3x_1^5 + w_1^2x_2^3x_1^5 + \\
&w_2x_2^4x_1^5 + w_1x_2^4x_1^5 + x_2^5x_1^5 + w_2^3x_2x_1^6 + w_2^2w_1x_2x_1^6 + w_2w_1^2x_2x_1^6 + w_1^3x_2x_1^6 + \\
&w_2x_2^3x_1^6 + w_1x_2^3x_1^6 + w_2^3x_1^7 + w_1^3x_1^7 + w_2^2x_2x_1^7 + w_2w_1x_2x_1^7 + w_1^2x_2x_1^7 + w_2x_2^2x_1^7 + \\
&w_1x_2^2x_1^7 + w_2^5w_1^2x_2^2 + w_2^4w_1^3x_2^2 + w_2^3w_1^4x_2^2 + w_2^2w_1^5x_2^2 + w_2^4w_1^2x_2^3 + w_2^3w_1^3x_2^3 + \\
&w_2^2w_1^4x_2^3 + w_2^4w_1x_2^4 + w_2w_1^4x_2^4 + w_2^3x_2^6 + w_1^3x_2^6 + w_2w_1x_2^7 + w_2^5w_1^2x_2x_1 + \\
&w_2^2w_1^5x_2x_1 + w_2^5w_1x_2^2x_1 + w_2^4w_1^2x_2^2x_1 + w_2^2w_1^4x_2^2x_1 + w_2w_1^5x_2^2x_1 + \\
&w_2^4w_1x_2^3x_1 + w_2^3w_1^2x_2^3x_1 + w_2^2w_1^3x_2^3x_1 + w_2w_1^4x_2^3x_1 + w_2^3x_2^5x_1 + w_1^3x_2^5x_1 + \\
&w_2^2x_2^6x_1 + w_2w_1x_2^6x_1 + w_1^2x_2^6x_1 + w_2x_2^7x_1 + w_1x_2^7x_1 + w_2^5w_1^2x_1^2 + w_2^4w_1^3x_1^2 + \\
&w_2^3w_1^4x_1^2 + w_2^2w_1^5x_1^2 + w_2^3w_1^3x_2x_1^2 + w_2^5x_2^2x_1^2 + w_2^4w_1^2x_2^2x_1^2 + w_2^3w_1^3x_2^2x_1^2 + \\
&w_2^2w_1^4x_2^2x_1^2 + w_2^4w_1x_2^3x_1^2 + w_2w_1^4x_2^3x_1^2 + w_2^3x_2^5x_1^2 + w_1^3x_2^5x_1^2 + \\
&w_2^2x_2^6x_1^2 + w_1^2x_2^6x_1^2 + x_2^7x_1^2 + w_2^5w_1x_1^3 + w_2^4w_1^2x_1^3 + w_2^3w_1^3x_1^3 + w_2^2w_1^4x_1^+ \\
&w_2w_1^5x_1^3 + w_2^4w_1x_2x_1^3 + w_2w_1^4x_2x_1^3 + w_2^3w_1x_2^2x_1^3 + w_2^2w_1^2x_2^2x_1^3 + \\
&w_2w_1^3x_2^2x_1^3 + w_2^2w_1x_2^3x_1^3 + w_2w_1^2x_2^3x_1^3 + w_2^5x_1^4 + w_2^4w_1x_1^4 + w_2^3w_1^2x_1^4 + \\
&w_2^2w_1^3x_1^4 + w_2w_1^4x_1^4 + w_1^5x_1^4 + w_2^4x_2x_1^4 + w_2^2w_1^2x_2x_1^4 + w_1^4x_2x_1^4 + w_2x_2^4x_1^4 + \\
&w_1x_2^4x_1^4 + x_2^5x_1^4 + w_2^4x_1^5 + w_1^4x_1^5 + w_2w_1x_2^2x_1^5 + x_2^4x_1^5 + w_2^3x_1^6 + w_1^3x_1^6 + \\
&w_2^2x_2x_1^6 + w_2w_1x_2x_1^6 + w_1^2x_2x_1^6 + w_2^2x_1^7 + w_1^2x_1^7 + w_2x_2x_1^7 + w_1x_2x_1^7 + w_2^5x_2^3 + \\
&w_2^4w_1x_2^3 + w_2w_1^4x_2^3 + w_2^5x_2^3 + w_2^2w_1^2x_2^4 + w_2^2w_1x_2^5 + w_2w_1^2x_2^5 + w_2^2x_2^6 + w_1^2x_2^6 + \\
&w_2x_2^7 + w_1x_2^7 + w_2^5w_1x_2x_1 + w_2w_1^5x_2x_1 + w_2^4w_1x_2^2x_1 + w_2w_1^4x_2^2x_1 + w_2^4x_2^3x_1 + \\
&w_2^2w_1^2x_2^3x_1 + w_1^4x_2^3x_1 + w_2^3x_2^4x_1 + w_1^3x_2^4x_1 + w_2w_1x_2^5x_1 + w_2^5w_1x_1^2 + w_2^4w_1^2x_1^2 + \\
&w_2^3w_1^3x_1^2 + w_2^2w_1^4x_1^2 + w_2w_1^5x_1^2 + w_2^5x_2x_1^2 + w_2^4w_1x_2x_1^2 + w_2w_1^4x_2x_1^2 + \\
&w_1^5x_2x_1^2 + w_2^2w_1^2x_2^2x_1^2 + w_2^3x_2^3x_1^2 + w_1^3x_2^3x_1^2 + w_2x_2^5x_1^2 + w_1x_2^5x_1^2 + \\
&w_2^5x_1^3 + w_2^4w_1x_1^3 + w_2w_1^4x_1^3 + w_1^5x_1^3 + w_2^4x_2x_1^3 + w_2^3w_1x_2x_1^3 + w_2^2w_1^2x_2x_1^3 + \\
&w_2^5x_2^3 + w_2^4w_1x_2^3 + w_2w_1^4x_2^3 + w_2^3x_2^2x_1^3 + w_2^2w_1x_2^2x_1^3 + w_2w_1^2x_2^2x_1^3 + w_1^3x_2^2x_1^3 + \\
&w_2^2x_2^3x_1^3 + w_2w_1x_2^3x_1^3 + w_1^2x_2^3x_1^3 + w_2^3w_1x_1^4 + w_2^2w_1^2x_1^4 + w_2w_1^3x_1^4 + \\
&w_2^3x_2x_1^4 + w_2^2w_1x_2x_1^4 + w_2w_1^2x_2x_1^4 + w_1^3x_2x_1^4 + w_2^2x_2^2x_1^4 + w_1^2x_2^2x_1^4 + \\
&w_2x_2^3x_1^4 + w_1x_2^3x_1^4 + w_2^3x_1^5 + w_2^2w_1x_1^5 + w_2w_1^2x_1^5 + w_1^3x_1^5 + w_2^2x_2x_1^5 + \\
&w_2w_1x_2x_1^5 + w_1^2x_2x_1^5 + w_2^2x_1^6 + w_1^2x_1^6 + w_2x_2x_1^6 + w_1x_2x_1^6 + x_2^2x_1^6 + \\
&w_2w_1x_1^6 + w_1^2x_1^6 + x_2^7 + w_2^5w_1 + w_2^4w_1^2 + w_2^3w_1^3 + w_2^2w_1^4 + w_2w_1^5 + w_2^5x_2 + w_2^4w_1x_2 + w_2^3w_1^2x_2 + \\
&w_2^2w_1^3x_2 + w_2w_1^4x_2 + w_1^5x_2 + w_2^2w_1^2x_2^2 + w_2^3x_2^3 + w_1^3x_2^3 + w_2^2x_2^4 + w_1^2x_2^4 + \\
&w_2x_2^5 + w_1x_2^5 + x_2^6 + w_2^5x_1 + w_2^3w_1^2x_1 + w_2^2w_1^3x_1 + w_1^5x_1 + w_2^2w_1^2x_2x_1 + w_2^2w_1x_2^2x_1 + \\
&w_2w_1^2x_2^2x_1 + w_2w_1x_2^3x_1 + w_2^2w_1^2x_1^2 + w_2^2x_2^2x_1^2 + w_2w_1x_2^2x_1^2 + w_1^2x_2^2x_1^2 + \\
&w_2^2w_1x_1^3 + w_2w_1^2x_1^3 + w_2^2x_2x_1^3 + w_1^2x_2x_1^3 + w_2x_2^2x_1^3 + w_1x_2^2x_1^3 + w_2x_2x_1^4 + \\
&w_1x_2x_1^4 + w_2x_1^5 + w_1x_1^5 + w_2^5 + w_1^5 + w_2^3w_1x_2 + w_2^2w_1^2x_2 + w_2w_1^3x_2 + w_2^3x_2^2 + w_1^3x_2^2 + \\
&w_2^2x_2^3 + w_2w_1x_2^3 + w_1^2x_2^3 + w_2x_2^4 + w_1x_2^4 + w_2^2w_1x_2x_1 + w_2w_1^2x_2x_1 + w_2^3x_1^2 + \\
&w_2^2w_1x_1^2 + w_2w_1^2x_1^2 + w_1^3x_1^2 + w_2w_1x_2x_1^2 + w_2w_1x_1^3 + x_2^2x_1^3 + x_1^5 + w_2^3w_1 + \\
&w_2^2w_1^2 + w_2w_1^3 + w_2^3x_2 + w_2^2w_1x_2 + w_2w_1^2x_2 + w_1^3x_2 + w_2w_1x_2^2 + w_2x_2^3 + w_1x_2^3 + x_2^4 + w_2^3x_1 + \\
&w_2^2w_1x_1 + w_2w_1^2x_1 + w_1^3x_1 + w_2^2x_2x_1 + w_2w_1x_2x_1 + w_1^2x_2x_1 + w_2x_2^2x_1 + w_1x_2^2x_1 + w_2^2x_1^2 + \\
&w_1^2x_1^2 + w_2x_2x_1^2 + w_1x_2x_1^2 + x_2^2x_1^2 + x_2x_1^3 + x_1^4 + w_2w_1x_2 + w_2^2x_1 + w_2w_1x_1 + w_1^2x_1 + \\
&x_2^2x_1 + w_2^2 + w_2w_1 + w_1^2 + x_1^2 + x_2
\end{aligned}
$$

Output with *noredSB*

$$
\begin{aligned}
J1[194] = {} & z_1 + w_2w_1x_2^3x_1^6 + x_2^5x_1^6 + w_2^3x_2x_1^7 + w_1^3x_2x_1^7 + w_2^4w_1^3x_2^3 + w_2^3w_1^4x_2^3 + w_2^5w_1x_2^4 + w_2^4w_1^2x_2^4 + \\
& w_2^2w_1^4x_2^4 + w_2w_1^5x_2^4 + w_2^3w_1^2x_2^5 + w_2^2w_1^3x_2^5 + w_2^4x_2^6 + w_2^3w_1x_2^6 + w_2^2w_1^2x_2^6 + w_2w_1^3x_2^6 + w_1^4x_2^6 + \\
& w_2^2w_1x_2^7 + w_2w_1^2x_2^7 + w_2^5w_1^2x_2^2x_1 + w_2^2w_1^5x_2^2x_1 + w_2^4w_1^2x_2^3x_1 + w_2^3w_1^3x_2^3x_1 + w_2^2w_1^4x_2^3x_1 + \\
& w_2^4w_1x_2^4x_1 + w_2w_1^4x_2^4x_1 + w_2^4x_2^5x_1 + w_2^2w_1^2x_2^5x_1 + w_1^4x_2^5x_1 + w_2^2w_1x_2^6x_1 + w_2w_1^2x_2^6x_1 + \\
& w_2^4w_1^3x_2x_1^2 + w_2^3w_1^4x_2x_1^2 + w_2^5w_1x_2^2x_1^2 + w_2w_1^5x_2^2x_1^2 + w_2^5x_2^3x_1^2 + \\
& w_2^4w_1x_2^3x_1^2 + w_2^3w_1^2x_2^3x_1^2 + w_2^2w_1^3x_2^3x_1^2 + w_2w_1^4x_2^3x_1^2 + w_1^5x_2^3x_1^2 + \\
& w_2^3w_1x_2^4x_1^2 + w_2w_1^3x_2^4x_1^2 + w_2^2w_1x_2^5x_1^2 + w_2w_1^2x_2^5x_1^2 + w_2^2x_2^6x_1^2 + w_2w_1x_2^6x_1^2 + \\
& w_1^2x_2^6x_1^2 + w_2x_2^7x_1^2 + w_1x_2^7x_1^2 + w_2^5w_1x_2x_1^3 + w_2^4w_1^2x_2x_1^3 + w_2^2w_1^4x_2x_1^3 + \\
& w_2w_1^5x_2x_1^3 + w_2^5x_2^2x_1^3 + w_1^5x_2^2x_1^3 + w_2^4x_2^3x_1^3 + w_1^4x_2^3x_1^3 + w_2w_1x_2^5x_1^3 + \\
& w_2x_2^6x_1^3 + w_1x_2^6x_1^3 + x_2^7x_1^3 + w_2^5w_1x_1^4 + w_2^4w_1^2x_1^4 + w_2^3w_1^3x_1^4 + w_2^2w_1^4x_1^4 + w_2w_1^5x_1^4 + w_2^5x_2x_1^4 + \\
& w_2^4w_1x_2x_1^4 + w_2w_1^4x_2x_1^4 + w_1^5x_2x_1^4 + w_2^4x_2^2x_1^4 + w_2^3w_1x_2^2x_1^4 + w_2w_1^3x_2^2x_1^4 + \\
& w_1^4x_2^2x_1^4 + w_2^3x_2^3x_1^4 + w_2^2w_1x_2^3x_1^4 + w_2w_1^2x_2^3x_1^4 + w_1^3x_2^3x_1^4 + w_2^2x_2^4x_1^4 + \\
& w_1^2x_2^4x_1^4 + w_2x_2^5x_1^4 + w_1x_2^5x_1^4 + w_2^2w_1^2x_2x_1^5 + w_2^2w_1x_2^2x_1^5 + w_2w_1^2x_2^2x_1^5 + \\
& w_2^2x_2^3x_1^5 + w_2w_1x_2^3x_1^5 + w_1^2x_2^3x_1^5 + w_2x_2^4x_1^5 + w_1x_2^4x_1^5 + x_2^5x_1^5 + w_2^3x_2x_1^6 + \\
& w_2^2w_1x_2x_1^6 + w_2w_1^2x_2x_1^6 + w_1^3x_2x_1^6 + w_2x_2^3x_1^6 + w_1x_2^3x_1^6 + w_2^3x_1^7 + w_1^3x_1^7 + \\
& w_2^2x_2x_1^7 + w_2w_1x_2x_1^7 + w_1^2x_2x_1^7 + w_2x_2^2x_1^7 + w_1x_2^2x_1^7 + w_2^5w_1^2x_2^2 + w_2^4w_1^3x_2^2 + \\
& w_2^3w_1^4x_2^2 + w_2^2w_1^5x_2^2 + w_2^4w_1^2x_2^3 + w_2^3w_1^3x_2^3 + w_2^2w_1^4x_2^3 + w_2^4w_1x_2^4 + \\
& w_2w_1^4x_2^4 + w_2^3x_2^6 + w_1^3x_2^6 + w_2w_1x_2^7 + w_2^5w_1^2x_2x_1 + w_2^2w_1^5x_2x_1 + w_2^5w_1x_2^2x_1 + \\
& w_2^4w_1^2x_2^2x_1 + w_2^2w_1^4x_2^2x_1 + w_2w_1^5x_2^2x_1 + w_2^4w_1x_2^3x_1 + w_2^3w_1^2x_2^3x_1 + w_2^2w_1^3x_2^3x_1 + \\
& w_2w_1^4x_2^3x_1 + w_2^3x_2^5x_1 + w_1^3x_2^5x_1 + w_2^2x_2^6x_1 + w_2w_1x_2^6x_1 + w_1^2x_2^6x_1 + w_2x_2^7x_1 + \\
& w_1x_2^7x_1 + w_2^5w_1^2x_1^2 + w_2^4w_1^3x_1^2 + w_2^3w_1^4x_1^2 + w_2^2w_1^5x_1^2 + w_2^3w_1^3x_2x_1^2 + \\
& w_2^5x_2^2x_1^2 + w_2^4w_1x_2^2x_1^2 + w_2w_1^4x_2^2x_1^2 + w_1^5x_2^2x_1^2 + w_2^4x_2^3x_1^2 + w_2^3w_1x_2^3x_1^2 + \\
& w_2^2w_1^2x_2^3x_1^2 + w_2w_1^3x_2^3x_1^2 + w_1^4x_2^3x_1^2 + w_2^2x_2^5x_1^2 + w_1^2x_2^5x_1^2 + x_2^7x_1^2 + w_2^5w_1x_1^3 + \\
& w_2^4w_1^2x_1^3 + w_2^3w_1^3x_1^3 + w_2^2w_1^4x_1^3 + w_2w_1^5x_1^3 + w_2^4w_1x_2x_1^3 + w_2w_1^4x_2x_1^3 + \\
& w_2^3w_1x_2^2x_1^3 + w_2^2w_1^2x_2^2x_1^3 + w_2w_1^3x_2^2x_1^3 + w_2^2w_1x_2^3x_1^3 + w_2w_1^2x_2^3x_1^3 + \\
& w_2^5x_2^4 + w_2^4w_1x_2^4 + w_2^3w_1^2x_2^4 + w_2^2w_1^3x_2^4 + w_2w_1^4x_2^4 + w_1^5x_2^4 + w_2^4x_2x_1^4 + \\
& w_2^2w_1^2x_2x_1^4 + w_1^4x_2x_1^4 + w_2x_2^4x_1^4 + w_1x_2^4x_1^4 + x_2^5x_1^4 + w_2^4x_1^5 + w_1^4x_1^5 + w_2w_1x_2^2x_1^5 + \\
& x_2^4x_1^5 + w_2^3x_1^6 + w_1^3x_1^6 + w_2^2x_2x_1^6 + w_2w_1x_2x_1^6 + w_1^2x_2x_1^6 + w_2^2x_1^7 + w_1^2x_1^7 + \\
& w_2x_2x_1^7 + w_1x_2x_1^7 + w_2^5x_2^3 + w_2^4w_1x_2^3 + w_2w_1^4x_2^3 + w_1^5x_2^3 + w_2^2w_1^2x_2^4 + w_2^2w_1x_2^5 + \\
& w_2w_1^2x_2^5 + w_2^2x_2^6 + w_1^2x_2^6 + w_2x_2^7 + w_1x_2^7 + w_2^5w_1x_2x_1 + w_2w_1^5x_2x_1 + \\
& w_2^4w_1x_2^2x_1 + w_2w_1^4x_2^2x_1 + w_2^4x_2^3x_1 + w_2^2w_1^2x_2^3x_1 + w_1^4x_2^3x_1 + w_2^2x_2^4x_1 + \\
& w_1^3x_2^4x_1 + w_2w_1x_2^5x_1 + w_2^5w_1x_1^2 + w_2^4w_1^2x_1^2 + w_2^2w_1^4x_1^2 + w_2^4x_2x_1^2 + w_2^3w_1^2x_2x_1^2 + \\
& w_2^5x_2^2x_1^2 + w_2^4w_1x_2^2x_1^2 + w_2w_1^4x_2^2x_1^2 + w_1^5x_2^2x_1^2 + w_2^4x_2^3x_1^2 + w_2^3w_1x_2^3x_1^2 + \\
& w_1^3x_2^3x_1^2 + w_2x_2^5x_1^2 + w_1x_2^5x_1^2 + w_2^5x_1^3 + w_2^4w_1x_1^3 + w_2w_1^4x_1^3 + w_1^5x_1^3 + \\
& w_2^4x_2x_1^3 + w_2^3w_1x_2x_1^3 + w_2^2w_1^2x_2x_1^3 + w_2w_1^3x_2x_1^3 + w_1^4x_2x_1^3 + w_2^3x_2^2x_1^3 + \\
& w_2^2w_1x_2^2x_1^3 + w_2w_1^2x_2^2x_1^3 + w_1^3x_2^2x_1^3 + w_2^2x_2^3x_1^3 + w_2w_1x_2^3x_1^3 + w_1^2x_2^3x_1^3 + \\
& w_2^3w_1x_1^4 + w_2^2w_1^2x_1^4 + w_2w_1^3x_1^4 + w_2^3x_2x_1^4 + w_2^2w_1x_2x_1^4 + w_2w_1^2x_2x_1^4 + w_1^3x_2x_1^4 + \\
& w_2^2x_2^2x_1^4 + w_1^2x_2^2x_1^4 + w_2x_2^3x_1^4 + w_1x_2^3x_1^4 + w_2^3x_1^5 + w_2^2w_1x_1^5 + w_2w_1^2x_1^5 + w_1^3x_1^5 + \\
& w_2w_1x_1^6 + w_2^5w_1^2 + w_2^4w_1^3 + w_2^3w_1^4 + w_2^2w_1^5 + w_2^4w_1^2x_2 + w_2^3w_1^3x_2 + w_2^2w_1^4x_2 + w_2^5x_2^2 + \\
& w_2^3w_1^2x_2^2 + w_2^2w_1^3x_2^2 + w_1^5x_2^2 + w_2^4x_2^3 + w_2^2w_1^2x_2^3 + w_1^4x_2^3 + w_2^2w_1x_2^4 + w_2w_1^2x_2^4 + \\
& w_2^2x_2^5 + w_2w_1x_2^5 + w_1^2x_2^5 + w_2^3w_1^2x_2x_1 + w_2^2w_1^3x_2x_1 + w_2^4x_2^2x_1 + w_1^4x_2^2x_1 + \\
& w_2^2x_2^4x_1 + w_2w_1x_2^4x_1 + w_1^2x_2^4x_1 + x_2^6x_1 + w_2^2w_1x_2^2x_1^2 + w_2w_1^2x_2^2x_1^2 + w_2^2x_2^3x_1^2 + \\
& w_1^2x_2^3x_1^2 + w_2^4x_1^3 + w_2^2w_1^2x_1^3 + w_1^4x_1^3 + w_2w_1x_2^2x_1^3 + w_2x_2^3x_1^3 + w_1x_2^3x_1^3 + x_2^4x_1^3 + \\
& w_2^3x_1^4 + w_1^3x_1^4 + w_2w_1x_2x_1^4 + w_2x_2^2x_1^4 + w_1x_2^2x_1^4 + w_2x_2^6 + w_1x_2^6 + x_2x_1^6 + x_1^7 + w_2^5w_1 + w_2^4w_1^2 + \\
& w_2^3w_1^3 + w_2^2w_1^4 + w_2w_1^5 + w_2^5x_2 + w_2^4w_1x_2 + w_2^3w_1^2x_2 + w_2^2w_1^3x_2 + w_2w_1^4x_2 + w_1^5x_2 + \\
& w_2^2w_1^2x_2^2 + w_2^3x_2^3 + w_1^3x_2^3 + w_2^2x_2^4 + w_1^2x_2^4 + w_2x_2^5 + w_1x_2^5 + x_2^6 + w_2^5x_1 + w_2^3w_1^2x_1 + \\
& w_2^2w_1^3x_1 + w_1^5x_1 + w_2^2w_1^2x_2x_1 + w_2^2w_1x_2^2x_1 + w_2w_1^2x_2^2x_1 + \\
& w_2w_1x_2^3x_1 + w_2^2w_1^2x_1^2 + w_2^2x_2^2x_1^2 + w_2w_1x_2^2x_1^2 + w_1^2x_2^2x_1^2 + w_2^2w_1x_1^3 + \\
& w_2w_1^2x_1^3 + w_2^2x_2x_1^3 + w_1^2x_2x_1^3 + w_2x_2^2x_1^3 + w_1x_2^2x_1^3 + w_2x_2x_1^4 + w_1x_2x_1^4 + w_2x_1^5 + w_1x_1^5 + \\
& w_2^5 + w_1^5 + w_2^3w_1x_2 + w_2^2w_1^2x_2 + w_2w_1^3x_2 + w_2^3x_2^2 + w_1^3x_2^2 + w_2^2x_2^3 + w_2w_1x_2^3 + w_1^2x_2^3 + \\
& w_2x_2^4 + w_1x_2^4 + w_2^2w_1x_2x_1 + w_2w_1^2x_2x_1 + w_2^3x_1^2 + w_2^2w_1x_1^2 + w_2w_1^2x_1^2 + w_1^3x_1^2 + \\
& w_2w_1x_2x_1^2 + w_2w_1x_1^3 + x_2^2x_1^3 + x_1^5 + w_2^3w_1 + w_2^2w_1^2 + w_2w_1^3 + \\
& w_2^3x_2 + w_2^2w_1x_2 + w_2w_1^2x_2 + w_1^3x_2 + w_2w_1x_2^2 + w_2x_2^3 + w_1x_2^3 + \\
& x_2^4 + w_2^3x_1 + w_2^2w_1x_1 + w_2w_1^2x_1 + w_1^3x_1 + w_2^2x_2x_1 + w_2w_1x_2x_1 + \\
& w_1^2x_2x_1 + w_2x_2^2x_1 + w_1x_2^2x_1 + w_2^2x_1^2 + w_1^2x_1^2 + w_2x_2x_1^2 + \\
& w_1x_2x_1^2 + x_2^2x_1^2 + x_2x_1^3 + x_1^4 + w_2w_1x_2 + w_2^2x_1 + w_2w_1x_1 + \\
& w_1^2x_1 + x_2^2x_1 + w_2^2 + w_2w_1 + w_1^2 + x_1^2 + x_2
\end{aligned}
$$