On decoding hyperbolic codes

Eduardo Camps-Moreno [1], Ignacio García-Marco [2], Hiram H. López [1], Irene Márquez-Corbella [2], Edgar Martínez-Moro [3], and Eliseo Sarmiento [4]

[1] Department of Mathematics, Virginia Tech, Blacksburg, VA, USA
{eduardoc, hhlopez}@vt.edu
[2] Facultad de Ciencias and Instituto de Matemáticas y Aplicaciones (IMAULL). Universidad de La Laguna, Spain
{iggarcia,imarquec}@ull.es
[3] Institute of Mathematics, University of Valladolid, Castilla Spain,
edgar.martinez@uva.es
[4] Instituto Politécnico Nacional, Mexico
esarmiento@ipn.mx

Abstract. This work studies several decoding algorithms for hyperbolic codes. We use some previous ideas to describe how to decode a hyperbolic code using the largest Reed-Muller code contained in it or using the smallest Reed-Muller code that contains it. A combination of these two algorithms is proposed when hyperbolic codes are defined by polynomials in two variables. Then, we compare hyperbolic codes and Cube codes (tensor product of Reed-Solomon codes) and propose decoding algorithms of hyperbolic codes based on their closest Cube codes. Finally, we adapt to hyperbolic codes the Geil and Matsumoto's generalization of Sudan's list decoding algorithm.

Keywords: Reed-Muller codes· evaluation codes · hyperbolic codes · decoding algorithms.

1 Introduction

Let $\mathbb{F}_q$ be a finite field with q elements. Given two vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$, the *Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$ is defined as $d_H(\mathbf{x}, \mathbf{y}) = |\{i \mid x_i \neq y_i\}|$, where $|\cdot|$ denotes the cardinality of the set. The *Hamming weight* of $\mathbf{x}$ is given by $\mathrm{w}_H(\mathbf{x}) = d_H(\mathbf{x}, \mathbf{0})$, where $\mathbf{0}$ denotes de zero vector in $\mathbb{F}_q^n$. The *support* of $\mathbf{x}$ is the set $\mathrm{supp}(\mathbf{x}) = \{i \mid x_i \neq 0\}$.

E. Camps-Moreno, H. H. López, E. Martínez-Moro and I. Márquez-Corbella were partially supported by Grant TED2021-130358B-I00 funded by MCIU/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR".
H. H. López was partially supported by the NSF grants DMS-2201094 and DMS-2401558.
I. García-Marco and I. Márquez-Corbella were partially supported by the Spanish MICINN PID2019-105896GB-I00

An $[n(C), k(C), \delta(C)]_q$ *linear code* C over $\mathbb{F}_q$ is an $\mathbb{F}_q$-vector space of $\mathbb{F}_q^{n(C)}$ with dimension $k(C)$, and minimum distance $\delta(C) = \min\{d_H(\mathbf{c}, \mathbf{c}') : \mathbf{c}, \mathbf{c}' \in C, \mathbf{c} \neq \mathbf{c}'\}$, thus its error correction capability is $t_{\mathcal{C}} = \left\lfloor \frac{\delta(C)-1}{2} \right\rfloor$. When there is no ambiguity, we write n, k, δ, and t instead of $n(C), k(C)$, $\delta(C)$, and $t_{\mathcal{C}}$, respectively.

Let $\mathbb{N}$ be the set of non-negative integers. For $A \subseteq \mathbb{N}^m$, $\mathbb{F}_q[A]$ is the $\mathbb{F}_q$-vector subspace of polynomials in $\mathbb{F}_q[\mathbf{X}] = \mathbb{F}_q[X_1, \ldots, X_m]$ with basis given by the set of monomials $\left\{\mathbf{X}^{\mathbf{i}} = X_1^{i_1} \cdots X_m^{i_m} \mid \mathbf{i} = (i_1, \ldots, i_m) \in A\right\}$. Let $\mathcal{P} = \mathbb{F}_q^m$, where $\mathcal{P} = \{P_1, \ldots, P_n\}$ and $n = |\mathcal{P}| = q^m$. Define the following evaluation map

$$\begin{array}{rccc} \mathrm{ev}_{\mathcal{P}} : \mathbb{F}_q[X_1, \ldots, X_m] & \longrightarrow & \mathbb{F}_q^n \\ f & \longmapsto & (f(P_1), \ldots, f(P_n)). \end{array}$$

The *monomial code associated* to A, denoted by $\mathcal{C}_A$, is defined as

$$\mathcal{C}_A = \mathrm{ev}_{\mathcal{P}}(\mathbb{F}_q[A]) = \{\mathrm{ev}_{\mathcal{P}}(f) \mid f \in \mathbb{F}_q[A]\}.$$

For $a, b \in \mathbb{R}$ and $a \leq b$, we denote by $[\![a, b]\!]$ the integer interval $[a, b] \cap \mathbb{Z}$.

Definition 1 (Reed-Muller and Hyperbolic codes).

- *Let $s \geq 0, m \geq 1$ be integers. The monomial code $\mathcal{C}_R$ where the set R is given by $R = \left\{\mathbf{i} = (i_1, \ldots, i_m) \in [\![0, q-1]\!]^m \mid \sum_{j=1}^m i_j \leq s\right\}$, is called the* Reed-Muller code *over $\mathbb{F}_q$ of degree s with m variables. This code is denoted by* $\mathrm{RM}_q(s, m)$.
- *Let $d, m \geq 1$ be integers. The monomial code $\mathcal{C}_H$ where the set H is given by $H = \left\{\mathbf{i} = (i_1, \ldots, i_m) \in [\![0, q-1]\!]^m \mid \prod_{j=1}^m (q - i_j) \geq d\right\}$, is called the* hyperbolic code *over $\mathbb{F}_q$ of order d with m variables. This code is denoted by* $\mathrm{Hyp}_q(d, m)$.

In our previous work [2] we proved there are two optimal Reed-Muller codes such that $\mathrm{RM}_q(s, m) \subseteq \mathrm{Hyp}_q(d, m) \subseteq \mathrm{RM}_q(s', m)$. In other words, the largest Reed-Muller code $\mathrm{RM}_q(s, m)$ contained in $\mathrm{Hyp}_q(d, m)$, and the smallest Reed-Muller code $\mathrm{RM}_q(s', m)$ that contains $\mathrm{Hyp}_q(d, m)$. We will use that result to propose several decoding procedures.

The paper is organized as follows. In Section 2, we describe two different algorithms to decode a hyperbolic code $\mathrm{Hyp}_q(d, m)$. These algorithms are based on the optimal Reed-Muller code that approximates to our hyperbolic code, that is, the largest Reed-Muller code $\mathrm{RM}_q(s, m)$ contained in $\mathrm{Hyp}_q(d, m)$, or the smallest Reed-Muller code $\mathrm{RM}_q(s', m)$ that contains $\mathrm{Hyp}_q(d, m)$. We will study the advantages and disadvantages in terms of efficiency and correction capability of these proposed algorithms. The choice of the algorithm to be used depends on which Reed-Muller code is closest to the hyperbolic code $\mathrm{Hyp}_q(d, m)$ as well as the efficiency or effectiveness that we need. At the end of that section, a third algorithm, which is a combination of the previous two, is adapted for $m = 2$, the case of two variables. In Section 3 a decoding algorithm for a hyperbolic code

$\mathrm{Hyp}_q(d,m)$ based on the tensor product of Reed-Solomon codes is presented. In Section 4 we recover Geil and Matsumoto's generalization of Sudan's List Decoding for order domain codes ([3]) but focus explicitly on hyperbolic codes. The novel idea that we present here is that we explicitly describe each of the sets involved in the algorithm or at least we give a subset of such sets. Finally in Section 5, we compare the performance of the five decoding algorithms proposed in this paper.

2 Decoding based on known Reed-Muller decoders

2.1 Decoding algorithm based on the smallest Reed-Muller code

The main idea that we present in this section is well-known and works for any pair of nested linear codes. Let $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{F}_q^n$ be two linear codes with parameters $[n,k_1,d_1]_q$ and $[n,k_2,d_2]_q$, respectively. Observe that a decoding algorithm for $\mathcal{C}_2$ that corrects up to t_2 errors is also a decoding algorithm for $\mathcal{C}_1$ that requires the same number of operations as in C_2, but corrects up to t_2 errors. Note that $d_2 \leq d_1$ and the difference between these two values might be huge. That is, a decoding algorithm for $\mathcal{C}_2$ is also a decoding algorithm for $\mathcal{C}_1$, but there is a loss in the number of errors that one might expect to correct.

Given a hyperbolic code $\mathrm{Hyp}_q(d,m)$, by [2, Theorem 3.6] we know the smallest integer s such that $\mathrm{Hyp}_q(d,m) \subseteq \mathrm{RM}_q(s,m)$. Thus, for each decoding algorithm for $\mathrm{RM}_q(s,m)$ we have one for $\mathrm{Hyp}_q(d,m)$. For example, if we use the already mentioned Pellikaan-Wu's list-decoding algorithm, one can correct up to $q^m(1-\sqrt{(q^m-\delta(\mathrm{RM}_q(s,m)))/q^m})$ errors.

Example 1. We have $\mathcal{C}_1 = \mathrm{Hyp}_9(9,2) \subseteq C_2 = \mathrm{RM}_9(s,2)$, where $s \geq 12$. Note that $\delta(C_2) = 5$, while $\delta(\mathcal{C}_1) = 9$. Using the algorithm explained in this section and Pellikkan-Wu's decoder for $\mathcal{C}_2$, we can correct up to 2 errors in $\mathcal{C}_1$ (which coincides with the error correcting capability of $\mathcal{C}_2$), while the error-correcting capability of the code $\mathcal{C}_1$ is $t_{\mathcal{C}_1} = 4$.

2.2 Decoding algorithm based on the largest Reed-Muller code

Take $A \subseteq B \subseteq \mathbb{N}^m$. We consider the set $\mathcal{Q} = \{f \in \mathbb{F}_q[\mathbf{X}] \mid \mathrm{supp}(f) \subseteq B \setminus A\}$, where the support of a polynomial $f \in \mathbb{F}_q[\mathbf{X}]$ is defined as

$$\mathrm{supp}(f) = \left\{ (i_1,\ldots,i_m) \mid f = \sum_{j=1}^{m} \alpha_{i_j}\mathbf{X}^{i_j}, \alpha_{i_j} \in \mathbb{F}_q \setminus \{0\} \right\}.$$

Observe that $|\mathcal{Q}| = q^{|B\setminus A|}$.

Let $\mathbf{y} \in \mathbb{F}_q^n$ be a received word. The following proposition tells us that, if the number of errors with respect to $\mathcal{C}_B$ is at most its error-correcting capability, i.e. $d_H(\mathbf{y},\mathcal{C}_B) \leq t_{\mathcal{C}_B}$, then, there exists a unique polynomial $f \in \mathcal{Q}$ such that the nearest codeword to $\mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f)$ is unique in $\mathcal{C}_A$.

Proposition 1. *Let $\mathbf{y} \in \mathbb{F}_q^n$ be a received word. Then there exists a polynomial $f \in \mathcal{Q}$ such that $d_H(\mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f), \mathcal{C}_A) = d_H(\mathbf{y}, \mathcal{C}_B)$. Moreover, if $d_H(\mathbf{y}, \mathcal{C}_B) \leq t_{\mathcal{C}_B}$, the polynomial f is unique.*

Proof. We first prove the existence of such polynomial. Set $t := d_H(\mathbf{y}, \mathcal{C}_B)$, then there exists a codeword $\mathbf{z} = \mathrm{ev}_{\mathcal{P}}(g) \in \mathcal{C}_B$, with $\mathrm{supp}(g) \subseteq B$, such that $d_H(\mathbf{y}, \mathbf{z}) = t$. Writing $g = f + \tilde{f}$, with $\mathrm{supp}(f) \subseteq B \setminus A$ and $\mathrm{supp}(\tilde{f}) \subseteq A$, we have $t = d_H(\mathbf{y}, \mathbf{z}) = d_H(\mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f), \mathrm{ev}_{\mathcal{P}}(\tilde{f}))$, with $f \in \mathcal{Q}$ and $\mathrm{ev}_{\mathcal{P}}(\tilde{f}) \in \mathcal{C}_A$.

Now we will prove the uniqueness when $t \leq t_{\mathcal{C}_B}$. Let $h \in \mathcal{Q}$ be another possible option. There exist two error-vectors $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{F}_q^n$ with weight smaller or equal to $t_{\mathcal{C}_B}$, such that $\mathrm{ev}_{\mathcal{P}}(f) + \mathbf{e}_1 = \mathrm{ev}_{\mathcal{P}}(h) + \mathbf{e}_2 = \mathbf{y}$. Therefore $\mathrm{ev}_{\mathcal{P}}(f) - \mathrm{ev}_{\mathcal{P}}(h) = \mathbf{e}_2 - \mathbf{e}_1$ is an element of $\mathcal{C}_B$, with weight at most $2t_{\mathcal{C}_B} < \delta(\mathcal{C}_B) - 1$. Thus, the difference $\mathrm{ev}_{\mathcal{P}}(f) - \mathrm{ev}_{\mathcal{P}}(h)$ must be zero and $f = h$. □

Let $\mathcal{C}_A \subseteq \mathcal{C}_B$ monomial codes such that $A \subseteq B \subseteq \mathbb{N}^m$. Let $\mathrm{Dec}_{\mathcal{C}_A}$ be a decoding algorithm for $\mathcal{C}_A$, which corrects up to E errors. By Proposition 1, we can define the following decoding algorithm for $\mathcal{C}_B$ that corrects also up to E errors.

Initialization: Let $\mathbf{y} \in \mathbb{F}_q^n$ be the received word. For each $f \in \mathcal{Q}$ we follow these steps
Step 1. Compute $\mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f)$.
Step 2. Decode using $\mathrm{Dec}_{\mathcal{C}_A}$ the word $\mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f)$.
Step 3. Denote by L the output list of **Step 2**, that is $L = \{\mathbf{c} \in \mathcal{C}_A \mid d_H(\mathbf{c}, \mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f)) \leq E\}$. If L is not empty, then for each $\mathbf{c}_A \in L$, we add to the output list $\mathbf{c}_A + \mathrm{ev}_{\mathcal{P}}(f)$.

The previous list-decoding algorithm for $\mathcal{C}_B$ corrects up to E errors and requires $q^{|B \setminus A|}$ calls to $\mathrm{Dec}_{\mathcal{C}_A}$. Moreover, if $E \leq t_{\mathcal{C}_B}$, one can easily transform the previous list-decoding algorithm into the following unique decoding one:

Initialization: Let $\mathbf{y} \in \mathbb{F}_q^n$ be the received word. Assume that the number of errors is at most $E \leq t_{\mathcal{C}_B}$.
Step 1 Compute $\mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f)$ for some $f \in \mathcal{Q}$.
Step 2 Decode using the decoder $\mathrm{Dec}_{\mathcal{C}_A}$ the word $\mathbf{y} - \mathrm{ev}_{\mathcal{P}}(f)$.
Step 3 If the result of **Step 2** is codeword $\mathbf{c}_A$ such that $\mathrm{w}_H(y - ev_P(f), \mathbf{c}_A) \leq t_{\mathcal{C}_B}$, then return $\mathbf{c}_A + \mathrm{ev}_{\mathcal{P}}(f)$.
Step 4 Otherwise, go back to **Step 1**.

The correctness of this algorithm is justified by Proposition 1 and the fact that $E \leq t_{\mathcal{C}_B} \leq t_{\mathcal{C}_A}$ and, hence, the output in **Step 2** has at most one element.

The algorithms proposed involve at most $|\mathcal{Q}| = q^{|B \setminus A|}$ calls to $\mathrm{Dec}_{\mathcal{C}_A}$, which could be interpreted as an inefficient algorithm from a theoretical point of view. Nevertheless, for practical purposes, if the difference between the sets B and A is small, this algorithm defines an efficient algorithm for $\mathcal{C}_B$ that corrects up to E errors, as long as an efficient algorithm for $\mathcal{C}_A$ exists and corrects the same number of errors.

2.3 Intermediate case.

In this section we are going to study an intermediate proposal between the two previous options. We will do our study for the case of two variables.

Proposition 2. *Let s be the smallest integer such that $\mathrm{Hyp}_q(d,2) \subseteq \mathrm{RM}_q(s,2)$. Let $H, R \subseteq \mathbb{N}^2$ such that $\mathcal{C}_H = \mathrm{Hyp}_q(d,2)$ and $\mathcal{C}_R = \mathrm{RM}_q(s-1,2)$. Then*

$$|H - R| \leq 2(\sqrt[4]{d} + 1).$$

Proof. We have that $\mathcal{C}_H = \mathrm{Hyp}_q(d,2) \subseteq \mathrm{RM}_q(s,2)$. An easy observation is that whenever $(i_1, i_2) \in H - R$, then $i_1 + i_2 = s$. Thus,

$$H - R = \{(t, s-t) \,|\, t \in [\![0, s]\!] \text{ and } (q-t)(q-s+t) \geq d\}. \tag{1}$$

Hence we are looking for those $t \in [\![0, s]\!]$ such that $(q-t)(q-s+t) \geq d$, or equivalently,

$$\left(q - \frac{s}{2}\right)^2 - \left(\frac{s}{2} - t\right)^2 \geq d \Rightarrow \left|\frac{s}{2} - t\right| \leq \underbrace{\sqrt{\left(q - \frac{s}{2}\right)^2 - d}}_{\Delta}.$$

If we compute Δ, then: $|H - R| \leq 2\Delta + 1$. We separate the proof in two cases depending on the parity of s. Recall that, by [2, Proposition 3.2], we know that $s = \left\lfloor 2(q - \sqrt{d}) \right\rfloor$.

1. If $s + 1 = 2r$, then $(r, r) \notin H$. As a consequence, $(q - \frac{s+1}{2})^2 < d$. Or equivalently, $\left(q - \frac{s}{2}\right)^2 - q + \frac{2s+1}{4} < d$. Thus,

$$\Delta \leq \sqrt{q - \frac{2q+1}{4}} < \sqrt{q - \frac{4(q - \sqrt{d}) - 1}{4}} = \sqrt{\sqrt{d} + \frac{1}{4}} < \sqrt[4]{d} + \frac{1}{2}, \text{ for } d \geq 1.$$

2. If $s = 2r$, then $(r, r+1) \notin H$. As a consequence, $\left(q - \frac{s}{2}\right)\left(q - \frac{s}{2} - 1\right) < d$. Or equivalently, $\left(q - \frac{s}{2}\right)^2 - \left(q - \frac{s}{2}\right) < d$. Thus,

$$\Delta \leq \sqrt{q - \frac{s}{2}} < \sqrt{q - \frac{2(q - \sqrt{d}) - q}{2}} = \sqrt{\sqrt{d} + \frac{1}{2}} < \sqrt[4]{d} + \frac{1}{2} \text{ for } d \geq 1.$$

□

Now we have the ingredients to define a new decoding algorithm for the code $\mathcal{C}_H = \mathrm{Hyp}_q(d,2)$. Let s be the smallest integer such that $\mathrm{Hyp}_q(d,2) \subseteq \mathrm{RM}_q(s,2)$. See [2, Proposition 3.2] for a precise description of such parameter s. We define $R \subseteq \mathbb{N}^2$ such that $\mathcal{C}_R = \mathrm{RM}_q(s-1,2)$. By Proposition 2 we know that $|H - R| \leq c\sqrt[4]{d}$. Let Dec_R be a decoding algorithm for $\mathcal{C}_R$ that corrects up to $t_R = \left\lfloor \frac{d(\mathcal{C}_R)-1}{2} \right\rfloor$ errors. Unifying the ideas of the above decoding algorithms (see sections 2.1 and 2.2), we have a decoding algorithm for $\mathcal{C}_H$ that corrects up

to E errors and requires $q^{c\sqrt[4]{d}}$ calls to Dec_R. If we compare it with the algorithm proposed in Section 2.1, we have a poorer complexity but we can correct more errors. This approach is particulary well suited when $d' = \delta(\mathrm{RM}_q(s,2)) \geq q$ and $|H - R|$ is small. In this case, $\delta(\mathcal{C}_R) = d' - q + 1$ and, as a consequence, the correction capability of the auxiliary Reed-Muller code we are using to decode is increased by around $q/2$.

Example 2. Consider $\mathcal{C} = \mathrm{Hyp}_q(d,2)$ with $q = 11$, and $d = 32$. See Figure 1 for a representation of this example. Throughout this example, we use Pellikaan-Wu's list-decoder (PW) for Reed-Muller codes. By [2, Proposition 3.2, Proposition 4.2] we have that $\mathrm{RM}_q(s',2) \subseteq \mathcal{C} \subseteq \mathrm{RM}_q(s,2)$ for $s' \leq 8$ and $s \geq 10$.

1. Thus, $\mathcal{C} \subseteq \mathcal{C}_{R_1} = \mathrm{RM}_q(10,2)$. Using Section 2.1 and applying PW algorithm on $\mathcal{C}_{R_1}$ we have an efficient decoding algorithm for $\mathcal{C}$ that **corrects up to** 5 **errors** and requires just one call to $\mathrm{Dec}_{\mathcal{C}_{R_1}}$.
2. Thus, $\mathcal{C}_{R_2} = \mathrm{RM}_q(8,2) \subseteq \mathcal{C}$. Using Section 2.2 to decode $\mathcal{C}$ and applying PW algorithm on $\mathcal{C}_{R_2}$ we have a decoding algorithm for $\mathcal{C}$ that **corrects up to** 16 **errors**. This algorithm uses the decoder $\mathrm{Dec}_{\mathcal{C}_{R_2}}$ plus some brute force. In particular, it requires q^{11} calls to $\mathrm{Dec}_{\mathcal{C}_{R_2}}$, where $11 = |H - R_2|$, and H and R_2 are the sets in $\mathbb{N}^2$ that define $\mathcal{C}$ and $\mathcal{C}_{R_2}$, respectively.
3. The intermediate proposal between the two previous options is described in Section 2.3. More precisely, we consider $\mathcal{C}_{R_3} = \mathrm{RM}_q(9,2)$ and we perform $q^{|H-R_3|} = q^5$ calls to the PW decoder for $\mathcal{C}_{R_3}$ to **correct up to** 10 **errors** in $\mathcal{C}$, where R_3 is the set in $\mathbb{N}^2$ that defines $\mathcal{C}_{R_3}$.

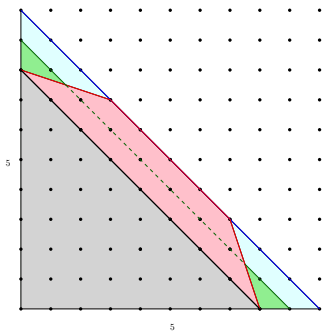

Fig. 1. Let $q = 11$ and $m = 2$. In this Figure the code $\mathrm{Hyp}_q(32,2)$ is equal to $\mathcal{C}_H$, $\mathrm{RM}_q(10,2)$ is equal to $\mathcal{C}_{R_1}$, $\mathrm{RM}_q(8,2)$ is equal to $\mathcal{C}_{R_2}$ and the code $\mathrm{RM}_q(9,2)$ equals to $\mathcal{C}_R$, where H, R_1, R_2 and R are the sets of lattice points below the red, the blue, the black and the green curve, respectively.

3 Decoding based on tensor products of RS codes

Let $f \in \mathbb{F}_q[X_1, \ldots, X_m]$ be a polynomial. The maximum degree of f with respect to X_i is denoted by $\deg_{X_i}(f)$. Now we define a family of monomial codes that we call cube codes of order s, which consist of the evaluation of the polynomials $f \in \mathbb{F}_q[X_1, \ldots, X_m]$ that satisfy that $\deg_{X_i}(f) \leq s$ for each $1 \leq i \leq m$, on the q^m points of $\mathbb{F}_q^m$. The formal definition is as follows.

Definition 2 (Cube codes). *Take $s \in \mathbb{N}$ and define $A = [\![0, s]\!]^m$. The monomial code $\mathcal{C}_A$, denoted by* $\mathrm{Cube}_q(s, m)$*, is called the* cube code *over $\mathbb{F}_q$ of order $s \geq 0$ with $m \geq 1$ variables.*

A Reed-Solomon code can be seen as a Reed-Muller code $\mathrm{RM}_q(d, m)$ of order d and $m = 1$ variables. That is, the Reed-Solomon code of order s is defined as

$$\mathrm{RS}_q(s) = \{\mathrm{ev}_{\mathcal{P}}(f) \mid f \in \mathbb{F}_q[X] \text{ and } \deg(f) \leq s\} = \mathrm{RM}_q(s, 1).$$

Reed-Solomon codes are one of the most popular and important families of codes. They are maximum distance separable (MDS) codes, thus a $\mathrm{RS}_q(s)$ is a code with parameters $[q, s+1, q-s]_q$. Reed-Solomon codes have efficient decoding algorithms. In the literature, the two primary decoding algorithms for Reed-Solomon codes are the Berlekamp-Massey algorithm [1], and the Sugiyama et al. adaptation of the Euclidean algorithm [4], both designed to solve a key equation.

Remark 1. Asume that the polynomials that define the Reed-Solomon code $\mathrm{RS}_q^i(s)$ belong to $\mathbb{F}_q[X_i]$. It is easy to see that the cube code $\mathrm{Cube}_q(s, m)$ is the tensor product of the m Reed-Solomon codes $\mathrm{RS}_q^1(s), \ldots, \mathrm{RS}_q^m(s)$. In other words, we have that

$$\mathrm{Cube}_q(s, m) = \mathrm{RS}_q^1(s) \otimes \cdots \otimes \mathrm{RS}_q^m(s).$$

Proposition 3. *The minimum distance of the cube code* $\mathrm{Cube}_q(s, m)$ *coincides with its footprint bound for the deglex monomial ordering. Therefore, the code $\mathcal{C} = \mathrm{Cube}_q(s, m)$ has length $n(\mathcal{C}) = q^m$, dimension $k(\mathcal{C}) = (s+1)^m$, and minimum distance $\delta(\mathcal{C}) = (q-s)^m$.*

Proof. This is a consequence of the fact that the cube code is the tensor product of Reed-Solomon codes. □

Proposition 4. *Take $d \in \mathbb{N}$. Then,*

(a) $\mathrm{Cube}_q(s, m) \subseteq \mathrm{Hyp}_q(d, m)$ *if and only if* $s \leq q - \sqrt[m]{d}$.

(b) $\mathrm{Hyp}_q(d, m) \subseteq \mathrm{Cube}_q(s', m)$ *if and only if* $s' \geq q - \left\lceil \frac{d}{q^{m-1}} \right\rceil$.

Proof. Let $H \subset [\![0, q-1]\!]^m$ such that $\mathcal{C}_H = \mathrm{Hyp}_q(d, m)$. If $\mathbf{i} = (i_1, \ldots, i_m)$ satisfies that $0 \leq i_j \leq q - \sqrt[m]{d}$ for all $1 \leq j \leq m$, then $\prod_{j=1}^m (q - i_j) \geq d$. Hence,

$\mathrm{Cube}_q(s,m) \subseteq \mathrm{Hyp}_q(d,m)$ for all $s \leq q - \sqrt[m]{d}$. If $s > q - \sqrt[m]{d}$, then $(q-s)^m < d$. Therefore $(s,\ldots,s) \notin H$ and $\mathrm{Cube}_q(s,m) \not\subseteq \mathrm{Hyp}_q(d,m)$.

We now check that $\mathrm{Hyp}_q(d,m) \subseteq \mathrm{Cube}_q(s',m)$ if and only if $s' \geq r := q - \left\lceil \frac{d}{q^{m-1}} \right\rceil$. For $s' \geq r$, it suffices to observe that for all $\mathbf{i} = (i_1,\ldots,i_m) \in H$, we have that $\max\{i_j\} \leq r$. Otherwise $\prod_{j=1}^{m}(q-i_j) \leq q^{m-1}(q-\max\{i_j\}) < q^{m-1}(q-r) \leq d$, a contradiction. Moreover, we have that $(r,0,\ldots,0) \in H$, so if $s' < r$ then $\mathrm{Hyp}_q(d,m) \not\subseteq \mathrm{Cube}_q(s',m)$. □

Theorem 1. *Let* $\mathrm{Dec}_{\mathrm{RS}}$ *be a decoding algorithm for* $\mathrm{RS}_q(s)$ *that corrects up to* t_{RS} *errors. Then, there exists a decoding algorithm* $\mathrm{Dec}_{\mathrm{Cube}_q(s,m)}$ *for* $\mathrm{Cube}_q(s,m)$ *that corrects up to* $(t_{\mathrm{RS}}+1)^m - 1$ *errors and requires calling* $f(m)$ *times the decoding algorithm* $\mathrm{Dec}_{\mathrm{RS}}$*, where*

$$f(m) = \sum_{i=0}^{m-1} (s+1)^{m-1-i} q^i \leq m q^{m-1} = \frac{mn}{q}, \text{ for all } m \geq 1.$$

Proof. We denote by $\alpha_1,\ldots,\alpha_q$ all the elements of $\mathbb{F}_q$. We proceed by induction on $m \in \mathbb{N}$. For $m = 1$, since $\mathrm{Cube}_q(s,1) = \mathrm{RS}_q(s)$ then, there exists $\mathrm{Dec}_{\mathrm{RS}}$ that corrects up to t_{RS} errors.

Now assume that there exists a decoding algorithm for $\mathrm{Cube}_q(s,m-1)$ that corrects up to $(t_{\mathrm{RS}}+1)^{m-1} - 1$ errors. Without loss of generality we reorder the points $\mathcal{P} = \{P_1,\ldots,P_n\} = \mathbb{F}_q^m$ with $n = q^m$ in such a way that the first q^{m-1} points of $\mathcal{P}$ are those that have α_1 in their first coordinate, then those that have α_2, and so on. Let $\mathbf{v} = (\mathbf{v}_1,\ldots,\mathbf{v}_q) \in \mathbb{F}_q^n$ where $\mathbf{v}_i \in \mathbb{F}_q^{q^{m-1}}$ be such that there exists $\mathbf{u} = (\mathbf{u}_1,\ldots,\mathbf{u}_q) \in \mathrm{Cube}_q(s,m)$ where $\mathbf{u}_i \in \mathbb{F}_q^{q^{m-1}}$ with $d(\mathbf{v},\mathbf{u}) < (t_{RS}+1)^m$. As $\mathbf{u} \in \mathrm{Cube}_q(s,m)$, there exists

$$f(X_1,\ldots,X_m) = \sum_{i_1,\ldots,i_m=0}^{s} \beta_{i_1,\ldots,i_m} X_1^{i_1}\cdots X_m^{i_m} \in \mathbb{F}_q[X_1,\ldots,X_m],$$

such that $\mathbf{u} = \mathrm{ev}_{\mathcal{P}}(f)$ and $\mathbf{u}_i = \mathrm{ev}_{\mathcal{P}'}(f(\alpha_i,X_2,\ldots,X_m))$ with $\mathcal{P}' = \mathbb{F}_q^{m-1}$. We are going to show how to recover $f(X_1,\ldots,X_m)$ from $\mathbf{v}$ by calling q times the decoder $\mathrm{Dec}_{\mathrm{Cube}_q(s,m-1)}$ and $(s+1)^{m-1}$ times the decoder $\mathrm{Dec}_{\mathrm{RS}}$.

For all $i \in \{1,\ldots,q\}$, we define $d_i = d(\mathbf{u}_i,\mathbf{v}_i)$. We say that $i \in \{1,\ldots,q\}$ is GOOD if $d_i < (t_{RS}+1)^{m-1}$; otherwise we say that i is BAD. Let d_{bad} be the number of BAD values $i \in \{1,\ldots,q\}$. Since

$$(t_{RS}+1)^m > \sum_{i=1}^{q} d_i \geq \sum_{i \text{ is BAD}} d_i \geq (t_{RS}+1)^{m-1} d_{bad},$$

we have that $d_{bad} < (t_{RS}+1)$.

Write $f(X_1,\ldots,X_m) = \sum_{j_2,\ldots,j_m=0}^{s} h_{j_2,\ldots,j_m}(X_1) X_2^{j_2}\cdots X_m^{j_m}$ where the univariate polynomial $h_{j_2,\ldots,j_m}(X_1) \in \mathbb{F}_q[X_1]$ has degree at most s, for all $j_2\ldots,j_m \in$

$\{0,\dots,s\}$. For all $\ell \in \{1,\dots,q\}$, consider

$$g_\ell(X_2,\dots,X_m) := f(\alpha_\ell, X_2,\dots,X_m) = \sum_{j_1,\dots,j_m=0}^{s} \beta_{j_1,\dots,j_m} {\alpha_\ell}^{j_1} X_2^{j_2}\cdots X_m^{j_m} =$$

$$= \sum_{j_2,\dots,j_m=0}^{s} h_{j_2,\dots,j_m}(\alpha_\ell) X_2^{j_2}\cdots X_m^{j_m} \in \mathbb{F}_q[X_2,\dots,X_m],$$

which satisfyies that $\deg_{X_i}(g_\ell) \leq s$ for all $i \in \{2,\dots,m\}$. Moreover, whenever ℓ is GOOD, we can recover g_ℓ by means of $\mathrm{Dec}_{\mathrm{Cube}_q(s,m-1)}$ and the values $\mathbf{v}_\ell = (v_{\ell_1},\dots,v_{\ell_{q^{m-1}}}) \in \mathbb{F}_q^{q^{m-1}}$.

Thus, if ℓ is GOOD, we have recovered the value $h_{j_2,\dots,j_m}(\alpha_\ell, X_2,\dots,X_m)$ for all $j_2,\dots,j_m \in \{0,\dots,s\}$. As there are at least $(q-t_{RS})$ GOOD values, then for each $j_2,\dots,j_m \in \{0,\dots,s\}$ we have at least $(q-t_{RS})$ correct evaluations of $h_{j_2,\dots,j_m}(X_1)$. Thus, by induction we can recover $\beta_{j_1,\dots,j_m}$ using $(s+1)^{m-1}$ times the decoder $\mathrm{Dec}_{\mathrm{RS}}$.

Now, let $f(m)$ be the number of times that algorithm $\mathrm{Dec}_{\mathrm{Cube}_q(s,m)}$ calls algorithm $\mathrm{Dec}_{\mathrm{RS}}$. We will deduce a formula for $f(m)$ by induction on $m \in \mathbb{N}$. First notice that for $m = 1$, since $\mathrm{Cube}_q(s,1) = \mathrm{RS}_q(s)$, we have that $f(1) = 1$. Moreover, from the above paragraphs, we can deduce that $f(m) = qf(m-1) + (s+1)^{m-1}f(1)$. Now we assume that $f(r) = qf(r-1) + (s+1)^{r-1}f(1)$ for all $r \leq m$ and we try to show the result for $f(m)$. Indeed,

$$\begin{aligned} f(m) &= qf(m-1) + (s+1)^{m-1}f(1) \\ &= q((s+1)^{m-2} + qf(m-2)) + (s+1)^{m-1} \\ &= (s+1)^{m-1} + q(s+1)^{m-2} + q^2 f(m-2) \\ &= \cdots = (s+1)^{m-1} + q(s+1)^{m-2} + q^2(s+1)^{m-3} + \cdots + q^{m-1}f(1) \\ &= \sum_{i=0}^{m-1} (s+1)^{m-1-i} q^i. \end{aligned}$$

This completes the proof. □

Example 3. Consider $\mathcal{C} = \mathrm{Hyp}_q(d,2)$ with $q = 32$, and $d = 225$. See Figure 2 for a representation of this example. We will give different decoding algorithms for $\mathcal{C}$ using all ideas proposed in this article so far.

1. By [2, Proposition 3.2], $\mathcal{C} \subseteq \mathrm{RM}_q(s,2)$, for $s \geq 34$. Therefore, using Section 2.1, and applying PW algorithm on $\mathcal{C}_{R_1} = \mathrm{RM}_q(34,2)$, we have an efficient decoding algorithm for $\mathcal{C}$ that **corrects up to** 11 **errors** and requires just one call to $\mathrm{Dec}_{\mathcal{C}_{R_1}}$.
2. By [2, Theorem 4.3], $\mathrm{RM}_q(s,2) \subseteq \mathcal{C}$, for $s \leq 24$. Therefore, using Section 2.2 and applying PW algorithm on $\mathcal{C}_{R_2} = \mathrm{RM}_q(24,2)$, we have a decoding algorithm for $\mathcal{C}$ that **corrects up to** 127 **errors** because the minimum distance of $RM_{32}(24,2)$ is 256 and so the error-correcting capability will be 127, which is beyond the error correction capability of $\mathcal{C}$. The algorithm uses

the decoder $\mathrm{Dec}_{\mathcal{C}_{R_2}}$ plus brute force. In particular it requires q^{156} calls to $\mathrm{Dec}_{\mathcal{C}_{R_2}}$ where $156 = |H - R_2|$, and H and R_2 are the sets in $\mathbb{N}^2$ defining $\mathcal{C}$ and $\mathcal{C}_{R_2}$, respectively.

3. The intermediate proposal between the two previous options described in Section 2.3 has a slight advantage in this example with respect to the first option. More precisely, one should consider $\mathcal{C}_{R'} = \mathrm{RM}_q(33, 2)$ and perform $q^{|H-R'|} = q$ calls to the PW decoder for $\mathcal{C}_{R'}$ to **correct up to** 14 **errors** because the minimum distance of $RM_{32}(33, 2)$ is 30 and so the error-correcting capability will be 14.
4. By Proposition 4.(b), we have $\mathcal{C} \subseteq \mathrm{Cube}_q(s, 2)$ for $s \geq 24$. We know an efficient decoding algorithm for $\mathcal{C}_3 = \mathrm{Cube}_q(24, 2)$ that corrects up to $t_3 = (t_{\mathrm{RS}} + 1)^m - 1 = 15$, where $t_{\mathrm{RS}} = \left\lfloor \frac{q-s-1}{2} \right\rfloor = 3$ denotes the error correction capability of $\mathrm{RS}_q(s)$ with $s = 24$. Therefore, using Theorem 1, we have an efficient decoding algorithm for $\mathcal{C}$ that **corrects up to** 15 **errors** and requires calling $q + s + 1 = 57$ times the decoder $\mathrm{Dec}_{\mathrm{RS}}$.

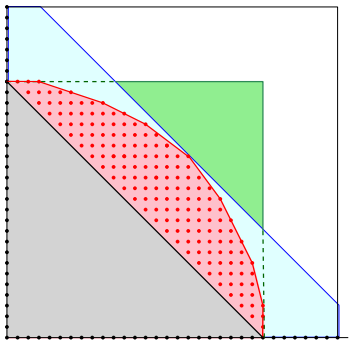

Fig. 2. Let $q = 32$ and $m = 2$. In this Figure the code $\mathrm{Hyp}_q(225, 2)$ is equal to $\mathcal{C}_H$, $\mathrm{RM}_q(24, 2)$ equals $\mathcal{C}_{R_1}$, the code $\mathrm{RM}_q(34, 2)$ is equal to $\mathcal{C}_{R_2}$ and the code $\mathrm{Cube}_q(24, 2)$ equals $\mathcal{C}_A$, where H, R_1, R_2 and A are the sets of lattice points below the red, the black, the blue and the green curve, respectively.

Remark 2. All the ideas proposed in Section 2 can be adapted to decode a hyperbolic code using a cube code. That is, by Proposition 4, given a hyperbolic code Hyp we can find the largest (respectively smallest) cube code contained in (respectively that contains) Hyp. And this result allow us to give decoding algorithms for hyperbolic codes in terms of the decoding algorithms of cube codes. Furthermore, an intermediate proposal between the two options can be used (as in Section 2.3).

4 Decoding based on a generalized Sudan's list decoding

Definition 3. *Let $\mathcal{C}_H = \mathrm{Hyp}_q(d,m)$ be a hyperbolic code with $H \subset [\![q-1]\!]^m$. For $r \geq 1$, take $H_r \subset [\![q-1]\!]^m$ such that $\mathcal{C}_{H_r} = \mathrm{Hyp}_q(r+1,m)$. For $i \geq 0$, we define*

$$L(d,r,i) = \{\mathbf{a} \in [\![q-1]\!]^m \mid \mathbf{a} + iH \subseteq H_r\}.$$

Observe that $L(d,r,0) = H_r$ for any d. We also have that $L(d,r,i+1) \subseteq L(d,r,i)$. Indeed, as $L(d,r,i+1)+(i+1)H \subset H_r$, then $L(d,r,i+1) \subset L(d,r,i+1)+H \subset L(d,r,i)$.

For the following algorithm, we assume that the numbers r and t satisfy the following conditions:

$$\sum_{i=0}^{\infty} \#L(d,r,i) > n \qquad \text{and} \qquad t = \min\left\{ i' \mid \sum_{i=0}^{i'} \#L(d,r,i) > n \right\}. \tag{2}$$

Notation: Given $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$, we define the Schur product as the component wise product on $\mathbb{F}_q^n$, i.e. $(\mathbf{u} * \mathbf{v})_i = u_i v_i$ and $(\mathbf{u}^{*j})_i = u_i^j$, for $j \geq 1$. We will write $\mathbf{u}^{*0}$ for the word with all the components equal to 1.
Initialization: Let $\mathbf{y} \in \mathbb{F}_q^n$ be the received word, and define r and t according to the conditions (2).

Step 1 For $0 \leq i \leq t$, find $Q_i \in \mathbb{F}_q[L(d,r,i)]$, not all zero, such that $\sum_{i=0}^{t} \mathrm{ev}(Q_i) * \mathbf{y}^{*i} = 0$.
Step 2 Factorize $Q(Y) = \sum_{i=0}^{t} Q_i Y^i \in \mathbb{F}_q[\mathbf{X}, Y]$ and detect all possible $f \in \mathbb{F}_q[\mathbf{X}]$ such that $(Y-f) | Q(Y)$. This can be done by the method of Wu [5].
Step 3 Return $\{\mathrm{ev}(f) \in \mathcal{C}_H \mid f \text{ is a solution of Step 2}\}$.

The list $\{\mathrm{ev}(f) \in \mathbb{F}_q[\mathbf{X}] \mid f \text{ is a solution of Step 2}\}$ is a list of at most t elements that contains all the codewords $\mathbf{x}$ in $\mathcal{C}_H$ such that $d_H(\mathbf{x},\mathbf{y}) \leq r$.

For completion, we write the proof of the last algorithm adapting it to the notation proposed in the previous lines.

Theorem 2. ([3, Theorem 4]) *The last algorithm gives the claimed output.*

Proof. Since $\sum_{i=0}^{t} \#L(d,r,i) > n$, then the equation $\sum_{i=0}^{t} \mathrm{ev}(Q_i) * \mathbf{y}^{*i} = 0$ has more indeterminates than equations and then a non-zero solution exists. Now, suppose that there exists $\mathbf{x} = \mathrm{ev}(f) \in \mathcal{C}_H$ such that $d_H(\mathbf{y},\mathbf{x}) \leq r$. Since $f \in H$, then all the monomials in the support of f^i are in iH. Then, all the monomials $X^{\mathbf{a}}$ appearing in $Q_i f^i$ satisfies $\mathbf{a} \in H_r$ by definition of $L(d,r,i)$. This implies that $\mathrm{ev}(Q(f)) \in \mathcal{C}_{H_r}$ and

$$\mathrm{w}_H(\mathrm{ev}(Q(f))) \geq r+1 \quad \text{or} \quad Q(f) = 0. \tag{3}$$

On the other hand, since $d_H(\mathbf{y},\mathbf{x}) \leq r$, we have that $\sum_{i=0}^{t} \mathrm{ev}(Q_i) * \mathbf{y}^{*i} = 0$ and $\mathrm{ev}(Q(f))$ can differ in at most r distinct positions. Thus $\mathrm{w}_H(\mathrm{ev}(Q(f))) \leq r$. By Equation (3) we conclude that $Q(f) = 0$, which means $(Y-f) | Q(Y)$. □

To use the previous algorithm and to know the number of errors that we can correct, we just need to compute $L(d, r, i)$ along with their sizes. In general, it is not clear what is the form of $L(d, r, i)$, but with the following results we can estimate their sizes, which is one of the main contributions of this section.

Proposition 5. *Let* $\mathcal{C}_A = \mathrm{Hyp}_q(d_A, m)$ *and* $\mathcal{C}_B = \mathrm{Hyp}_q(d_B, m)$. *Then* $\mathcal{C}_{A+B} \subset \mathrm{Hyp}_q(d_A + d_B - q^m, m)$.

Proof. Take $f \in \mathbb{F}_q[A]$ and $g \in \mathbb{F}_q[B]$. Denote the set of zeros of f (resp. g) in $\mathbb{F}_q^m$ by $Z(f)$ (resp. $Z(g)$). We know that $|Z(f)| \leq q^m - d_A$ and $|Z(g)| \leq q^m - d_B$. This implies that $|Z(fg)| \leq |Z(f)| + |Z(g)| \leq 2q^m - d_A - d_B$. In other words, for any $fg \in \mathcal{C}_{A+B}$, we have that $\mathrm{w}_H(\mathrm{ev}(f) * \mathrm{ev}(g)) \geq d_A + d_B - q^m$. Then $\delta(\mathcal{C}_{A+B}) \geq d_A + d_B - q^m$. As $\mathcal{C}_{A+B}$ is a monomial code, its minimum distance is the minimum of the footprints of its defining monomials. Thus we obtain the conclusion. □

With the above result we can bound the size of the set $L(d, r, 1)$ and so, we can bound the number of errors that we can correct with the proposed algorithm when we adapt it to unique decoding.

Corollary 1. *Let* $\mathcal{C} = \mathrm{Hyp}_q(d, m)$ *and* $d \geq r \geq 1$. *If* $\mathcal{C}_{H_1} = \mathrm{Hyp}_q(q^m + r - d + 1, m)$, *then* $H_1 \subseteq L(d, r, 1)$.

The number of errors that we can uniquely decode with the proposed algorithm is given by an easy-to-check formula.

Corollary 2. *If* $\#H_r + \#H_1 > n$, *then the algorithm can correct up to* r *errors solving a linear equation in* $(\mathbb{F}_q[\mathbf{X}])[Y]$.

If $m = 2$ we can do even better. In the case of two variables, we can not only bound the size of the set $L(d, r, 1)$ but we can also know exactly what monomial code is associated with such subset.

Proposition 6. *Let* $\mathcal{C}_H = \mathrm{Hyp}_q(d, 2)$, *with* $d > q$. *Take* $a = \left\lfloor q - \frac{d}{q} \right\rfloor$ *and* $b = q - \frac{d}{q-a}$. *For* $r < a - b + 1$, *we have* $\mathcal{C}_{L(d,r,1)} = \mathrm{Cube}_q(q - 1 - a, 2)$.

Proof. Take $c = q - 1 - a$. Observe that $(0, a) \in H$ but for any $a < i_2 \in \mathbb{Z}$, $(0, i_2) \notin H$. Similarly, $(q - a)(q - b) \geq d$ but for any $i_1 > b$, $(q - a)(q - i_1) < d$. As

$$(q - c - a)(q - c - b) = a - b + 1 > r,$$

and since $\{(i_1, i_2) \mid i_1 + i_2 \leq a + b\}$ and H_r are both convex sets, then for any $(i_1, i_2) \in H$ such that $i_1 + i_2 \leq a + b$, we have $(i_1 + c, i_2 + c) \in H_r$.

Now, suppose that $(i_1, i_2) \in H$ but $i_1 + i_2 > a + b$. Then we have

$$\begin{aligned}
(q - i_1 - c)(q - i_2 - c) &= (q - i_1)(q - i_2) + c(i_1 + i_2 - 2q) + c^2 \\
&\geq (q - a)(q - b) + c(i_1 + i_2 - 2q) + c^2 \\
&> (q - a)(q - b) + c(a + b - 2q) + c^2 \\
&= (q - a - c)(q - b - c) \\
&= a - b + 1 \\
&> r.
\end{aligned}$$

This means that $(i_1+c, i_2+c) \in H_r$. Then we have $c+H \subset H_r$. By Definition 3, we can easily see that if (i_1, i_2) satisfies the property that $i_1, i_2 \leq c$, then $(i_1, i_2) \in L(d, r, 1)$.

Finally, we can assure $L(d, r, 1) = \{(i_1, i_2) \mid i_1, i_2 \leq c\}$, otherwise it would exist some $(i_1, i_2) \in L(d, r, 1)$ with $i_1 > c$ such that $i_1 + a > q - 1$ and $(i_1 + c, i_2 + c) \in H_r$, which is a contradiction. □

The last result can be generalized for all the sets $L(d, r, i)$.

Corollary 3. *Let $\mathcal{C}_H = \mathrm{Hyp}_q(d, 2)$, $d > q$, and a, b and r as before. Then $\mathcal{C}_{L(d,r,i)} = \mathrm{Cube}_q(q - 1 - ia, 2)$.*

Proof. We know the case $i = 1$. Assume the result is true for $i \in \mathbb{N}$. As $L(d, r, i+1) + H \subseteq L(d, r, i) = \mathrm{Cube}_q(q - 1 - ia, 2)$ and $\mathrm{Hyp}_q(d, 2) \subseteq \mathrm{Cube}_q(a, 2)$ [2, Theorem 4.3], then we have that for any $(i_1, i_2) \in H$, $i_1, i_2 \leq a$. This implies that

$$(q - 1 - (i + 1)a + i_1, q - 1 - (i + 1)a + i_2) \leq (q - 1 - ia, q - 1 - ia).$$

We conclude that $\{(i_1, i_2) \mid i_1, i_2 \leq q-1-(i+1)a\} \subseteq L(d, r, i+1)$. The equality follows from the fact that $(a, 0)$ is a point in H. □

Remark 3. Using the previous results, for $m = 2$ we have the following bound for the number of errors that our algorithm uniquely corrects. Take $t = \left\lfloor \frac{q-1}{a} \right\rfloor$ and $r < a-b+1$, with a and b as before for $\mathcal{C}_H = \mathrm{Hyp}_q(d, 2)$. If $\#H_r + \sum_{i=1}^{t}(q - 1 - ia)^2 > n$, then the algorithm can correct up to r errors.

Example 4. Consider $\mathcal{C} = \mathrm{Hyp}_q(d, 2)$ with $q = 16$ and $d = 81$. See Figure 3 for a representation of this example. Take $r = 8$. Brute force computation on Geil and Matsumoto's algorithm gives that:

- $C_{L(d,r,0)} = \mathrm{Hyp}_q(r + 1, 2)$, which coincide with Definition 3.
- Moreover, $C_{L(d,r,1)} = \mathrm{Cube}_q(5, 2)$, which matches with Proposition 6 since

$$\mathrm{Cube}_q(5, 2) = \mathrm{Cube}_q(q - 1 - a, 2) \text{ with } a = \left\lfloor q - \frac{d}{q} \right\rfloor = 10.$$

- Finally, $C_{L(d,r,2)} = \{0\}$.

Moreover, following Remark 3, $r = 8$ is the maximum number of errors that we can correct with this algorithm since

$$r < a - b + 1, \text{ where } a = \left\lfloor q - \frac{d}{q} \right\rfloor = 10 \quad \text{and} \quad b = q - \frac{d}{q - a} = 2.5.$$

Observe that $\#H_r + \sum_{i=1}^{t}(q - 1 - ia)^2 = \#L(d, r, 0) + \#L(d, r, 1) > n = 256$, where $t = \left\lfloor \frac{q-1}{a} \right\rfloor = 1$ and $L(d, r, 0)$ and $L(d, r, 1)$ are the set of lattice points below the green and the black curve of Figure 3, respectively. The sets $L(d, r, 0)$ and $L(d, r, 1)$ are, in general, difficult to describe. Using the results of this section we explicitly describe these sets when $m = 2$, and we provide a lower bound on their sizes for the cases when $m > 2$.

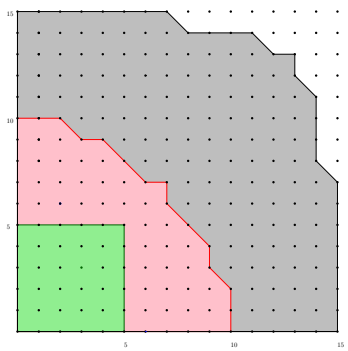

Fig. 3. Let $q = 16$ and $m = 2$. In this Figure the code $\mathcal{C} = \mathrm{Hyp}_q(81, 2)$ is equal to $\mathcal{C}_H$, $\mathcal{C}_{L(81,8,0)} = \mathrm{Hyp}_q(9, 2)$ is equal to $\mathcal{C}_{L_0}$ and $\mathcal{C}_{L(81,8,1)} = \mathrm{Cube}_q(5, 2)$ is equal to $\mathcal{C}_{L_1}$ where H, L_0 and L_1 are the sets of lattice points below the red, the black and the green curve, respectively.

5 Comparisons and Conclusions

Table 1 compares the performance of the five decoding algorithms proposed in this paper for the hyperbolic code $\mathcal{C} = \mathrm{Hyp}_q(d, m)$, where $q = 32, m = 2$ and d takes different values. Table 1 is composed by 6 blocks, one for each value of d. Each block contains 5 lines, which represent the following:

- First line refers to the algorithm of Section 2.1. Here, we compute the smallest integer s such that $\mathcal{C} \subseteq \mathrm{RM}_q(s, m)$. Then, we use Pellikaan-Wu list-decoding algorithm for Reed-Muller codes to decode $\mathcal{C}$.
- Second line refers to the algorithm of Section 2.2. Here, we compute the largest integer s such that $\mathrm{RM}_q(s, m) \subseteq \mathcal{C}$. Then, we use the Pellikaan-Wu list-decoding algorithm for Reed-Muller codes to decode $\mathcal{C}$ plus some brute force.
- Third line refers to the algorithm of Section 2.3, an intermediate case between the above two options. In this case, we use again the Pellikaan-Wu list-decoding algorithm for Reed-Muller codes to decode $\mathcal{C}$.
- Fourth line refers to the algorithm of Section 3. More precisely, we compute the smaller integer s such that $\mathcal{C} \subseteq \mathrm{Cube}_q(s, m)$. Then, we use the algorithm described in Section 3 for cube codes to decode $\mathcal{C}$.
- Fifth line refers to the specific algorithm known for $\mathcal{C}$ described in Section 4.

The third column describes the number of calls to the corresponding decoder. The last column represents the minimum distance of the auxiliary code that we are using in each case.

In Table 1 we observe that the algorithm with the greatest error correcting capability is always achieved by the method given in the second line. However, the huge amount of calls to the decoder makes it highly impractical. The third

method always corrects more errors than the first one, but requires more calls to the decoder. Concerning the third, fourth and fifth method, we find instances where each of the methods outperforms the others. All the algorithms we propose except the fifth one, rely on a known decoder for either a Reed-Muller or a cube code. As a consequence, a better decoder for any of these codes would imply better error correction capability. Interestingly, when decoding in terms of cube codes, we reduce many times the problem to a code with a promisingly high minimum distance, but then we use a decoding algorithm with poor error correcting capability. We consider that it is an interesting problem to find better decoding algorithms for cube codes which, in particular, will lead to better decoding algorithms for hyperbolic codes.

References

1. E. Berlekamp, *Non-binary BCH Decoding*. IEEE Transactions on Information Theory, 14(2), 242, 1968.
2. E. Camps, H. López, I. García-Marco, I. Márquez-Corbella, E. Martínez-Moro and E. Sarmiento *On the generalized Hamming weights of hyperbolic codes*. Journal of Algebra and its Applications. Volume 23, Issue 07, 2024. Article number 2550062
3. O. Geil, and R. Matsumoto. *Generalized Sudan's list decoding for order domain codes*. In International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, pp. 50-59. Springer, Berlin, Heidelberg, 2007.
4. Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. *A method for solving key equation for decoding Goppa codes*. Information and Control, 27, pp. 87–99, 1975.
5. X.-W.Wu, *An Algorithm for Finding the Roots of the Polynomials Over Order Domains*. In: 2002 IEEE International Symposium on Information Theory, p. 202. IEEE Press, New York, 2002.

	Error correcting capability	Number of calls	Called algorithm	Minimum distance
$d = 257$	$E_1 = 16$	1	$\mathrm{Dec}(\mathrm{RM}_q(31,2))$	$\delta(\mathrm{RM}_q(31,2)) = 32$
	$E_2 = 155$	q^{134}	$\mathrm{Dec}(\mathrm{RM}_q(23,2))$	$\delta(\mathrm{RM}_q(23,2)) = 288$
	$E_3 = 32$	q^8	$\mathrm{Dec}(\mathrm{RM}_q(30,2))$	$\delta(\mathrm{RM}_q(30,2)) = 64$
	$E_4 = 24$	56	$\mathrm{Dec}(\mathrm{RS}_q(23))$	$\delta(\mathrm{Cube}_q(24,2)) = 81$
	$E_5 = 23$	1	$\mathrm{Dec}(\mathrm{Hyp}_q(d,2))$	$\delta(\mathrm{Hyp}_q(d,2)) = 257$
$d = 225$	$E_1 = 14$	1	$\mathrm{Dec}(\mathrm{RM}_q(34,2))$	$\delta(\mathrm{RM}_q(34,2)) = 29$
	$E_2 = 137$	q^{156}	$\mathrm{Dec}(\mathrm{RM}_q(24,2))$	$\delta(\mathrm{RM}_q(24,2)) = 256$
	$E_3 = 15$	q	$\mathrm{Dec}(\mathrm{RM}_q(33,2))$	$\delta(\mathrm{RM}_q(33,2)) = 30$
	$E_4 = 15$	57	$\mathrm{Dec}(\mathrm{RS}_q(24))$	$\delta(\mathrm{Cube}_q(24,2)) = 64$
	$E_5 = 19$	1	$\mathrm{Dec}(\mathrm{Hyp}_q(d,2))$	$\delta(\mathrm{Hyp}_q(d,2)) = 225$
$d = 193$	$E_1 = 13$	1	$\mathrm{Dec}(\mathrm{RM}_q(36,2))$	$\delta(\mathrm{RM}_q(36,2)) = 27$
	$E_2 = 118$	q^{182}	$\mathrm{Dec}(\mathrm{RM}_q(25,2))$	$\delta(\mathrm{RM}_q(25,2)) = 224$
	$E_3 = 14$	q^3	$\mathrm{Dec}(\mathrm{RM}_q(35,2))$	$\delta(\mathrm{RM}_q(35,2)) = 28$
	$E_4 = 15$	58	$\mathrm{Dec}(\mathrm{Cube}_q(25,2))$	$\delta(\mathrm{Cube}_q(25,2)) = 49$
	$E_5 = 15$	1	$\mathrm{Dec}(\mathrm{Hyp}_q(d,2))$	$\delta(\mathrm{Hyp}_q(d,2)) = 193$
$d = 150$	$E_1 = 12$	1	$\mathrm{Dec}(\mathrm{RM}_q(39,2))$	$\delta(\mathrm{RM}_q(39,2)) = 24$
	$E_2 = 83$	q^{212}	$\mathrm{Dec}(\mathrm{RM}_q(27,2))$	$\delta(\mathrm{RM}_q(27,2)) = 160$
	$E_3 = 12$	q^6	$\mathrm{Dec}(\mathrm{RM}_q(38,2))$	$\delta(\mathrm{RM}_q(38,2)) = 25$
	$E_4 = 8$	60	$\mathrm{Dec}(\mathrm{RS}_q(27))$	$\delta(\mathrm{Cube}_q(27,2)) = 25$
	$E_5 = 9$	1	$\mathrm{Dec}(\mathrm{Hyp}_q(d,2))$	$\delta(\mathrm{Hyp}_q(d,2)) = 150$
$d = 65$	$E_1 = 8$	1	$\mathrm{Dec}(\mathrm{RM}_q(47,2))$	$\delta(\mathrm{RM}_q(47,2)) = 16$
	$E_2 = 49$	q^{343}	$\mathrm{Dec}(\mathrm{RM}_q(29,2))$	$\delta(\mathrm{RM}_q(29,2)) = 96$
	$E_3 = 8$	q^6	$\mathrm{Dec}(\mathrm{RM}_q(46,2))$	$\delta(\mathrm{RM}_q(46,2)) = 17$
	$E_4 = 3$	62	$\mathrm{Dec}(\mathrm{RS}_q(29))$	$\delta(\mathrm{Cube}_q(29,2)) = 9$
	$E_5 = 5$	1	$\mathrm{Dec}(\mathrm{Hyp}_q(d,2))$	$\delta(\mathrm{Hyp}_q(d,2)) = 65$
$d = 15$	$E_1 = 3$	1	$\mathrm{Dec}(\mathrm{RM}_q(56,2))$	$\delta(\mathrm{RM}_q(56,2)) = 7$
	$E_2 = 7$	q^{64}	$\mathrm{Dec}(\mathrm{RM}_q(48,2))$	$\delta(\mathrm{RM}_q(48,2)) = 15$
	$E_3 = 4$	q^3	$\mathrm{Dec}(\mathrm{RM}_q(55,2))$	$\delta(\mathrm{RM}_q(55,2)) = 8$
	$E_4 = 0$	64	$\mathrm{Dec}(\mathrm{RS}_q(31))$	$\delta(\mathrm{Cube}_q(31,2)) = 1$
	$E_5 = 0$	1	$\mathrm{Dec}(\mathrm{Hyp}_q(d,2))$	$\delta(\mathrm{Hyp}_q(d,2)) = 15$

Table 1. Comparison between the five different algorithms described above to decode $\mathrm{Hyp}_{32}(d,2)$, for different values of d.