arXiv:1107.2321v1 [math.NT] 12 Jul 2011

# An algorithm for list decoding number field codes

Jean-François Biasse[1] and Guillaume Quintin[2]

[1] Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4
biasse@lix.polytechnique.fr
[2] École Polytechnique, 91128 Palaiseau, France
quintin@lix.polytechnique.fr

**Abstract.** We present an algorithm for list decoding codewords of algebraic number field codes in polynomial time. This is the first explicit procedure for decoding number field codes whose construction were previously described by Lenstra [14] and Guruswami [9]. We rely on an equivalent of the LLL reduction algorithm for $\mathcal{O}_K$-modules due to Fieker and Stehlé [6] and on algorithms due to Cohen [3] for computing the Hermite normal form of matrices representing modules over Dedekind domains.

## 1 Introduction

Algorithms for list decoding Reed-Solomon codes, and their generalization the algebraic-geometric codes are now well understood. The codewords consist of sets of functions whose evaluation at a certain number of points are sent, thus allowing the receiver to retrieve them provided that the number of errors is manageable. Some algebraic-geometric codes beat the Gilbert-Varshamov bound for alphabet sizes $q > 49$ (see [7, 18]), and possess a nice algebraic structure which enables to decode even in the presence of a large number of errors [12, 17].

The idea behind behind algebraic-geometric codes can be adapted to define algebraic codes whose messages are encoded as a list of residues redundant enough to allow errors during the transmission. The Chinese Remainder codes (CRT codes) have been fairly studied by the community [11, 15]. The encoded messages are residues modulo $N := p_1, \cdots, p_n$ of numbers $m \leq K := p_1 \cdots p_k$ where $p_1 < p_2 < \cdots < p_n$ are prime numbers. They are encoded by using

$$\begin{aligned} \mathbb{Z} &\longrightarrow \quad \mathbb{Z}/p_1 \times \cdots \times \mathbb{Z}/p_n \\ m &\longmapsto (m \bmod p_1, \cdots, m \bmod p_n). \end{aligned}$$

Decoding algorithms for CRT codes were significantly improved to reach the same level of tolerance to errors as those for Reed-Solomon codes [2, 8, 11]. As algebraic-geometric codes are a generalization of Reed-Solomon codes, the idea arose that we could generalize the results for CRT codes to redundant residue codes based on number fields. Indeed, we can easily define an analogue of the

CRT codes where a number field $K$ plays the role of $\mathbb{Q}$ and its ring of integers $\mathcal{O}_K$ plays the role of $\mathbb{Z}$. Then, for prime ideals $\mathfrak{p}_1, \cdots, \mathfrak{p}_n$ such that $\mathcal{N}(\mathfrak{p}_1) < \cdots < \mathcal{N}(\mathfrak{p}_n)$, a message $m \in \mathcal{O}_K$ can be encoded by using

$$\begin{array}{rcl} \mathcal{O}_K & \longrightarrow & \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n \\ c: \ m & \longmapsto & (m \bmod \mathfrak{p}_1, \cdots, m \bmod \mathfrak{p}_n). \end{array}$$

The construction of good codes on number fields have been independantly studied by Lenstra [14] and Guruswami [9]. They provided indications on how to chose number fields having good properties for the underlying codes, but they did not describe any decoding algorithm.

**Contribution:** The main contribution of this paper is to provide the first algorithm for decoding number field codes. We adapt an algorithm of Guruswami [10, Chap. 7] for CRT codes, and present original contributions to methods for manipulating modules over the ring of integers of a number field (see Section 7). Throughout the article, we denote by $K$ a number field of degree $d$, of discriminant $\Delta$ and of ring of integers $\mathcal{O}_K$. The prime ideals $(\mathfrak{p}_i)_{i \leq n}$ satisfy $\mathcal{N}(\mathfrak{p}_1) < \mathcal{N}(\mathfrak{p}_2) < \cdots < \mathcal{N}(\mathfrak{p}_n)$, and we define $N := \prod_{i \leq n} \mathcal{N}(\mathfrak{p}_i)$ and $B := \prod_{i \leq k} \mathcal{N}(\mathfrak{p}_i)$ for integers $k, n$ such that $0 < k < n$. Before describing our algorithm in more details in the following sections, let us state the main result of the paper.

**Theorem.** *Let $\varepsilon > 0$, and a message $m \in \mathcal{O}_K$ satisfying $\|m\| \leq B$, then there is an algorithm that returns all the messages $m' \in \mathcal{O}_K$ such that $\|m'\| \leq B$ and that $c(m)$ and $c(m')$ have mutual agreement $t$ satisfying*

$$t \geq \sqrt{k(n+\varepsilon)}.$$

*This algorithm is polynomial in $d$ , $\log(N)$, $1/\varepsilon$ and $\log|\Delta|$.*

## 2 Previous work

Even though little has been done on decoding algorithms for number field codes, their construction is not recent. Indeed, they were described in the 1980's by Lenstra [14], and in 2003, Guruswami [9] introduced classes of number field codes that correspond to the description we gave in the introduction. The main difference between the two approaches is the fact that Lenstra used the residue of the message at both Archimedian and non-Archimedian places whereas the approach of Guruswami (that we followed here) consists of restricting ourselves to Archimedian places. In fact, the approach of Lenstra allows to prove the existence of good asymptotic codes more easily, but it is then harder to pursue the analogy with Reed-Solomon and CRT codes. Nevertheless, Guruswami [9] exhibited a code family $\{\mathcal{C}_i\}$ of increasing block length $n_i \to \infty$ such that

$$\liminf \frac{k_i}{n_i} > 0 \quad \text{and} \quad \liminf \frac{d_i}{n_i} > 0,$$

where $k_i$ denotes the dimension of $\mathcal{C}_i$ and $d_i$ denotes its minimum distance.

In his thesis, Guruswami [10, Chap. 7] described a more general framework embracing both algebraic-geometric codes and CRT codes. The messages lie in a ring $R$, and they are encoded via a $n$-uplet of residues modulo coprime ideals $I_1, \cdots, I_n$ of $R$ by using the function

$$\begin{array}{rl} R & \longrightarrow \qquad R/I_1 \times \cdots \times R/I_n \\ c : m & \longmapsto (m \bmod I_1, \cdots, m \bmod I_n). \end{array}$$

Then, the messages with sufficient agreement are recovered as roots of a polynomial in $R[y]$. In [10], this general approach is applied to CRT codes, as well as to Reed-Solomon codes and algebraic-geometric codes. Finding a suitable polynomial of degree $l$ for decoding boils down to finding short elements in a sub $R$-module of $R^{l+1}$. This is achieved in [10, Chap. 7] for CRT codes by the means of the LLL algorithm [13]. In this paper, we use an equivalent of this algorithm on $\mathcal{O}_K$-modules of $K^{l+1}$ described by Fieker and Stehlé [6], and we provide a proper analysis of the construction of the module containing our decoding polynomial, thus allowing to follow the approach of the general framework [10, Chap. 7]. We use the same notations as in [10, Chap. 7] as much as possible, however, the proof of many results need to be adjusted to our context, although the general strategy remains unchanged. As we see in Section 7, the major technical difference is that we need to manipulate $\mathcal{O}_K$-modules defined by their pseudo-basis.

Note that Cohn and Heninger [5] recently gave an ideal form of Coppersmith's theorem that can be applied to enhance list decoding of Reed-Solomon and Algebraic-geometric codes. They used a weak version of the result of Fieker and Stehlé [6] to give a variant of this theorem for number fields. They discussed its applications to the shortest vector problem in integral lattices in the context of lattice-based cryptography, but they did not consider the possibility to list decode number field codes.

## 3 Generalities on number fields

Let $K$ be a number field of degree $d$. It has $r_1 \leq d$ real embeddings $(\theta_i)_{i \leq r_1}$ and $2r_2$ complex embeddings $(\theta_i)_{r_1 < i \leq 2r_2}$ (coming as $r_2$ pairs of conjugates). The field $K$ is isomorphic to $\mathcal{O}_K \otimes \mathbb{Q}$ where $\mathcal{O}_K$ denotes the ring of integers of $K$. We can embed $K$ in

$$K_{\mathbb{R}} := K \otimes \mathbb{R} \simeq \mathbb{R}^{r_1} \times \mathbb{C}^{r_2},$$

and extend the $\theta_i$'s to $K_{\mathbb{R}}$. Let $T_2$ be the Hermitian form on $K_{\mathbb{R}}$ defined by

$$T_2(x, x') := \sum_i \theta_i(x)\overline{\theta_i}(x'),$$

and let $\|x\| := \sqrt{T_2(x, x)}$ be the corresponding $L_2$-norm. Let $(\alpha_i)_{i \leq d}$ such that $\mathcal{O}_K = \oplus_i \mathbb{Z}\alpha_i$, then the discriminant of $K$ is given by $\Delta = \det^2(T_2(\alpha_i, \alpha_j))$. The norm of an element $x \in K$ is defined by $\mathcal{N}(x) = \prod_i |\theta_i(x)|$.

We encode our messages with prime ideals of $\mathcal{O}_K$. However, for decoding, we need a more general notion of ideal, namely the fractional ideals of $\mathcal{O}_K$. They can be defined as finitely generated $\mathcal{O}_K$-modules of $K$. When a fractional ideal is contained in $\mathcal{O}_K$, we refer to it as an integral ideal, which is in fact an ideal of $\mathcal{O}_K$. For every fractional ideal $I$ of $\mathcal{O}_K$, there exists $r \in \mathbb{Z}$ such that $rI$ is integral. The sum and product of two fractional ideals of $\mathcal{O}_K$ is given by

$$IJ = \{i_1 j_1 + \cdots + i_l j_l \mid l \in \mathbb{N}, i_1, \cdots i_l \in I, j_1, \cdots j_l \in J\}$$
$$I + J = \{i + j \mid i \in I, j \in J\}.$$

The fractional ideals of $\mathcal{O}_K$ are invertible, that is for every fractional ideal $I$, there exists $I^{-1} := \{x \in K \mid xI \subseteq \mathcal{O}_K\}$ such that $II^{-1} = \mathcal{O}_K$. The set of fractional ideals is equipped with a norm function defined by $\mathcal{N}(I) = \det(I)/\det(\mathcal{O}_K)$. The norm of ideals is multiplicative, and in the case of an integral ideal, we have $\mathcal{N}(I) = |\mathcal{O}_K/I|$. Also note that the norm of $x \in K$ is precisely the norm of the principal ideal $(x) = x\mathcal{O}_K$. Algorithms for ideal arithmetic in polynomial time in the bit size of $\Delta$ are extensively described in [4].

In the following, we will study finitely generated sub $\mathcal{O}_K$-module of $\mathcal{O}_K[y]$. Let $M \subseteq K^l$ be a finitely generated $\mathcal{O}_K$-module. As in [3, Chap. 1], we say that $[(a_i), (\mathfrak{a}_i)]_{i \leq n}$, where $a_i \in K$ and $\mathfrak{a}_i$ is a fractional ideal, is a pseudo-basis for $M$ if

$$M = \mathfrak{a}_1 a_1 \oplus \cdots \oplus \mathfrak{a}_n a_n.$$

Note that a pseudo-basis is not unique, and the main result of [6] is precisely to compute a pseudo-basis of short elements. If the sum is not direct, we call $[(a_i), (\mathfrak{a}_i)]_{i \leq n}$ a pseudo-generating set for $M$. Once a pseudo-generating set $[(a_i), (\mathfrak{a}_i)]_{i \leq n}$ for $M$ is known, we can associate a pseudo-matrix $A = (A, I)$ to $M$, where $A \in K^{n \times l}$ and $I = (\mathfrak{a}_i)_{i \leq n}$ is a list of $n$ fractional ideals such that

$$M = \mathfrak{a}_1 A_1 + \cdots + \mathfrak{a}_n A_n,$$

where $A_i \in K^l$ is the $i$-th row of $A$. We can construct a pseudo-basis from a pseudo-generating set by using the Hermite normal form (HNF) over Dedekind domains (see [3, Th. 1.4.6]). Assume $A$ is of rank $l$ (in particular $n \geq l$), then there exists a $n \times n$ matrix $U = (u_{i,j})$ and $n$ non-zero ideals $\mathfrak{b}_1, \cdots, \mathfrak{b}_n$ satisfying

1. $\forall i, j, u_{i,j} \in \mathfrak{b}_i^{-1}\mathfrak{a}_j$.
2. $\mathfrak{a} = \det(U)\mathfrak{b}$ for $\mathfrak{a} = \prod_i \mathfrak{a}_i$ and $\mathfrak{b} = \prod_i \mathfrak{b}_i$.
3. The matrix $UA$ is of the form

$$UA = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \vdots & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ * & * & \dots & 1 \\ \hdashline & & (0) & \end{pmatrix}.$$

4. $M = \mathfrak{b}_1\omega_1 \oplus \cdots \oplus \mathfrak{b}_l\omega_l$ where $\omega_1, \cdots \omega_l$ are the first $l$ rows of $UA$.

In general, the algorithm of [3] for computing the HNF of a pseudo-matrix takes exponential time, but as in the integer case, there exists a modular one which is polynomial in the dimensions of $A$, the degree of $K$, and the bit size of the modulo. Note that in the case of a pseudo matrix representing an $\mathcal{O}_K$-module $M$, the modulo an integral multiple of the determinantal ideal $\mathfrak{g}(M)$, which is generated by all the ideals of the form

$$\det_{i_1,\cdots,i_l}(A) \cdot \mathfrak{a}_{i_1} \cdots \mathfrak{a}_{i_l},$$

where $\det_{i_1,\cdots,i_l}(A)$ is the determinant of the $l \times l$ minor consisting of the rows of indices $i_1, \cdots, i_l$. The determinantal ideal is a rather involved structure, except in the case $l = n$.

## 4 Johnson-type bound for number fields codes

A Johnson-type bound is a positive number $J$ depending on the distance, the blocklength and the cardinalities of the Alphabets constituting the code. It garanties that a "small" number of codewords are in any sphere of radius $J$. By "small" number, we mean a number of codewords which is linear in the code blocklength and the cardinality of the code. In our case, the Johnson-type bound for number fields codes depends only on the code blocklength and its minimal distance, and "small" means polynomial in $\sum_{i=1}^{n} \log \mathcal{N}(\mathfrak{p}_i)$.

The Johnson-type bound of [10, Section 7.6.1] remains valid for number field codes. For any prime ideal $\mathfrak{p} \subset \mathcal{O}_K$, the quotient $\mathcal{O}_K/\mathfrak{p}$ is a finite field. Thus the $i$'th symbol of a codeword comes from an alphabet of size $\mathcal{N}(\mathfrak{p}_i) = |\mathcal{O}_K/\mathfrak{p}_i|$ and [10, Th. 7.10] can be applied. Let $t$ be the least positive integer such that

$$\prod_{i=1}^{t} \mathcal{N}(\mathfrak{p}_i) > \left(\frac{2B}{d}\right)^d,$$

where $d = [K : \mathbb{Q}]$ and let

$$T = \prod_{i=1}^{t} \mathcal{N}(\mathfrak{p}_i).$$

Then, by [9, Lem. 12], the minimal hamming distance of the number fields code is at least $n - t + 1$. Using [10, Th. 7.10], we can show that for a given message and $\varepsilon > 0$, only a "small" number of codewords satisfy

$$\sum_{i=1}^{n} a_i > \sqrt{(t+\varepsilon)n}, \tag{1}$$

where $a_i = 1$ if the codeword and the message agree at the $i$-th position, $a_i = 0$ otherwise. Thus, if our list decoding algorithm returns all the codewords having

at most $n-\sqrt{(t+\varepsilon)n}$ errors then this number is garanteed to be "small". Therefore, the Johnson bound appears to be a good objective for our algorithm. Note that we would derive a different bound by using weighted distances. In particular, by using the log-weighted hamming distance i.e. $d(x,y) = \sum_{i:x\neq y \mod \mathfrak{p}_i} \log \mathcal{N}(\mathfrak{p}_i)$, the condition would be $\sum_{i=1}^{n} a_i \log \mathcal{N}(\mathfrak{p}_i) > \sqrt{(\log T + \varepsilon)\log N}$.

## 5 General description of the algorithm

In this section, we give a high-level description of our decoding algorithm. We follow the approach of the general framework described in [10], making the arrangements required in our context. Our code is the set of $m \in \mathcal{O}_K$ such that $\|m\| \leq B$ where $B = \prod_{i\leq k} \mathcal{N}(\mathfrak{p}_i)$. We also define $N := \prod_{i\leq n} \mathcal{N}(\mathfrak{p}_i)$. A codeword $m$ is encoded via

$$\begin{array}{rcl} \mathcal{O}_K & \longrightarrow & \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n \\ m & \longmapsto & (m \bmod \mathfrak{p}_1, \cdots, m \bmod \mathfrak{p}_n). \end{array}$$

Let $z_1, \cdots, z_n$ be non-negative real numbers, and let $Z$ be a parameter. In this section, as well as in Section 6 and 7, we assume that the $z_i$ are integers. We assume that we received a vector $r_1, \cdots, r_n \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$. We wish to retrieve all the codewords $m$ such that $\sum_i a_i z_i > Z$ where $a_i = 1$ if $m \bmod \mathfrak{p}_i = r_i$ and 0 otherwise (we say that $m$ and $(r_i)_{i\leq n}$ have weighted agreement $Z$).

We find the codewords $m$ with desired weighted agreement by computing roots of a polynomial $c \in \mathcal{O}_K[y]$ that satisfies

$$\|m\| \leq B \Longrightarrow \|c(m)\| < F, \tag{2}$$

for an appropriate bound $F$. A polynomial $c$ satisfying (2) is chosen in the ideal $\prod_{i\leq n} J_i^{z_i} \subseteq \mathcal{O}_K[y]$ where

$$J_i = \{a(y)(y - r_i) + p \cdot b(y) \mid a, b \in \mathcal{O}_K[y] \text{ and } p \in \mathfrak{p}_i\}.$$

With such a choice of a polynomial, we necessarily have $c(m) \in \prod_i \mathfrak{p}_i^{z_i a_i}$, where $a_i = 1$ if $c(m) \bmod \mathfrak{p}_i = r_i$, 0 otherwise. In particular, if $c(m) \neq 0$ this means that $\mathcal{N}(c(m)) \geq \prod_i \mathcal{N}(\mathfrak{p}_i)^{z_i a_i}$. In addition, we know from the arithmetic-geometric inequality that $\|c(m)\| \geq \sqrt{d}\mathcal{N}(c(m))^{1/d}$. We thus know that if the weighted agreement satisfies

$$\sum_{i\leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) > -\frac{d}{2}\log(d) + d\log(F), \tag{3}$$

which in turns implies $\sqrt{d}\left(\prod_i \mathcal{N}(\mathfrak{p}_i)^{z_i a_i}\right)^{1/d} > F$, then $c(m)$ has to be zero, since otherwise it would contradict (2).

**Algorithm 1** Decoding algorithm

**Input:** $\mathcal{O}_K$, $z_1, \cdots, z_n$, $B$, $Z$, $r_1, \cdots, r_n \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$.
**Output:** All $m$ such that $\sum_i a_i z_i > Z$.
1: Compute $l$ and $F$.
2: Find $c \in \prod_{i \leq n} J_i^{z_i} \subseteq \mathcal{O}_K[y]$ of degree at most $l$ such that $\|m\| \leq B \Longrightarrow \|c(m)\| < F$.
3: Find all roots of $c$ and report those roots $\xi$ such that $\|\xi\| \leq B$ and $\sum_i a_i z_i > Z$.

## 6 Existence of the decoding polynomial

In this section, we prove the existence of a polynomial $c \in \mathcal{O}_K[y]$ and a constant $F > 0$ such that for all $\|m\| \leq B$, $m \in \mathcal{O}_K$, we have $\|c(m)\| \leq F$. This proof is not constructive. The actual computation of this polynomial will be described in Section 7. We first need to estimate the number of elements of $\mathcal{O}_K$ bounded by a given size.

**Lemma 1.** *Let $F' > 0$ and $0 < \gamma < 1$, then the number of $x \in \mathcal{O}_K$ such that $\|x\| \leq F'$ is at least*

$$\left\lfloor \frac{\pi^{d/2} F'^d}{2^{r_1+r_2-1+\gamma}\sqrt{|\Delta|}\Gamma(d/2)} \right\rfloor.$$

*Proof.* As in [16, Chap. 5], we use the standard results of Minkowski theory for our purposes. More precisely, there is an isomorphism

$$f : K_{\mathbb{R}} \longrightarrow \mathbb{R}^{r_1+2r_2}$$

and a scalar product $(x, y) := \sum_{i \leq r_1} x_i y_i + \sum_{r_1 < i \leq r_2} 2x_i y_i$ on $\mathbb{R}^{r_1+2r_2}$ transfering the canonical measure from $K_{\mathbb{R}}$ to $\mathbb{R}^{r_1+2r_2}$. Let $\lambda = f(\mathcal{O}_K)$, $X := \{x \in K_{\mathbb{R}} \mid \|x\| \leq F'\}$, and $m \in \mathbb{N}$. We know from Minkowski's lattice point theorem that if

$$\mathrm{Vol}(X) > m 2^d \det(\lambda),$$

then $\#(f(x) \cap \lambda) \geq m$. As $\mathrm{Vol}(X) = 2^{r_2}\left(2\pi^{d/2} F'^d / \Gamma(d/2)\right)$ and $\det(\lambda) = \sqrt{|\Delta|}$, we have the desired result.

Then, we must derive from Lemma 1 an analogue of [10, Lemma 7.6] in our context. This lemma allows us to estimate the number of polynomials of degree $l$ satisfying (2). To simplify the expressions, we use the following notation in the rest of the paper

$$\alpha_{d,\Delta,\gamma} := \frac{\pi^{d/2}}{2^{r_1+r2-1+\gamma}\sqrt{|\Delta|}\Gamma(d/2)}.$$

**Lemma 2.** *For positive integers $B, F'$, the number of polynomial $c \in \mathcal{O}_K[y]$ of degree at most $l$ satisfying (2) is at least*

$$\left(\alpha_{d,\Delta,\gamma}\left(\frac{F'}{(l+1)B^{l/2}}\right)^d\right)^{l+1}.$$

*Proof.* Let $c(y) = c_0 + c_1 y + \cdots + c_l y^l$. We want the $c_i$'s to satisfy $\|c_i m^i\| < F'/(l+1)$ whenever $\|m\| \leq B$. This is the case when $\|c_i\| < F'/(B^i(l+1))$. By Lemma 1, there are at least $\alpha_{d,\Delta,\gamma}\left(F'/((l+1)B^i)\right)^d$ possibilities for $c_i$. Therefore, the number of polynomial $c$ satisfying (2) is at least

$$(\alpha_{d,\Delta,\gamma})^{l+1}\left(\left(\frac{F'}{l+1}\right)^{l+1}\prod_{i=0}^{l}B^{-i}\right)^d,$$

which finishes the proof.

Now that we know how to estimate the number of $c \in \mathcal{O}_K[y]$ or degree at most $l$ satisfying (2), we need to find a lower bound on $F$ to ensure that we can find such a polynomial in $\prod_i J_i^{z_i}$. The following lemma is an equivalent of [10, Lemma 7.7].

**Lemma 3.** *Let $l, B, F$ be positive integers, there exists $c \in \prod_i J_i^{z_i}$ satisfying* (2) *provided that*

$$F > 2(l+1)B^{l/2}\frac{1}{(\alpha_{d,\Delta,\gamma})^{1/d}}\left(\prod_i \mathcal{N}(\mathfrak{p}_i)^{\binom{z_i+1}{2}}\right)^{\frac{1}{d(l+1)}}. \tag{4}$$

*Proof.* Let us apply Lemma 2 to $F' = F/2$. There are at least

$$\left(\alpha_{d,\Delta,\gamma}\left(\frac{F/2}{(l+1)B^{l/2}}\right)^d\right)^{l+1}$$

polynomial $c \in \mathcal{O}_K[y]$ satisfying $\|m\| \leq B \Rightarrow \|c(m)\| < F/2$. In addition, we know from [10, Corollary 7.5] that $\prod_i |\mathcal{N}(\mathfrak{p}_i)|^{\binom{z_i+1}{2}} \geq |\mathcal{O}_K[y]/\prod_i J_i^{z_i}|$, which implies that if (4) is satisfied, then necessarily

$$\left(\alpha_{d,\Delta,\gamma}\left(\frac{F/2}{(l+1)B^{l/2}}\right)^d\right)^{l+1} > \left|\mathcal{O}_K[y]/\prod_i J_i^{z_i}\right|.$$

This means that there are at least two distinct polynomials $c_1, c_2 \in \mathcal{O}_K[y]$ of degree at most $l$ such that $(c_1 - c_2) \in \prod_i J_i^{z_i}$ and $\|c_1(m)\|, \|c_2(m)\| < F/2$ whenever $\|m\| \leq B$. The choice of $c := c_1 - c_2$ finishes the proof.

## 7 Computation of the decoding polynomial

Let $l > 0$ be an integer to be determined later. To compute $c \in \prod_i J_i^{z_i}$ of degree at most $l$ satisfying (2), we need to find a short pseudo-basis of the sub $\mathcal{O}_K$-module $M$ of $K^{l+1}$ given by

$$c = c_0 + yc_1 + \cdots + y^l c_l \longrightarrow (c_0, c_1, \cdots, c_l).$$

We first compute a peudo-generating set for each $J_i^{z_i}$, then we compute a pseudo-basis for their intersection, and we finally call the algorithm of [6] to produce a short peudo-basis of $M$ from which we derive $c$.

An algorithm for computing a pseudo-basis of the intersection of two modules given by their pseudo basis is described by Cohen in [3, 1.5.2]. Applying this successively for each pseudo-basis of $J_i^{z_i}$ would certainly produce a pseudo-basis of $\prod_i J_i^{z_i}$, but it would be likely to be exponential. We present in the following an algorithm for computing the intersection of $n$ sub $\mathcal{O}_K$-modules of $K^{l+1}$ that is polynomial in $n$, $l$ and the bit size of the product of the $n$ determinantal ideals when the modules are included in $\mathcal{O}_K^{l+1}$.

**Algorithm 2** Intersection of $\mathcal{O}_K$-modules

**Input:** $\mathcal{O}_K$-modules $M_1, \cdots, M_n$ given by $(A_1, (\mathfrak{a}_i^{(1)})_{i\le l+1}), \cdots, (A_n, (\mathfrak{a}_i^{(n)})_{i\le l+1})$.
**Output:** A pseudo matrix $(H, (\mathfrak{c}_i)_{i\le l+1})$ representing $\cap_i M_i$

1: $\mathfrak{g} \leftarrow \mathfrak{g}(M_1)\cdots\mathfrak{g}(M_n)$, $H_1 \leftarrow A_1$, $(\mathfrak{c}_i)_{i\le l+1} \leftarrow (\mathfrak{a}_i^1)_{i\le l+1}$.
2: **for** $2 \le k \le n$ **do**
3: Let $B_k$ be the pseudo matrix given by

$$B_k = \left(\begin{array}{c:cc} A_k & A_k & (0) \\ \hdashline (0) & & H_{k-1} \end{array}\right),$$

and the ideals $(\mathfrak{d}_i)_{i\le k(l+1)} := \mathfrak{a}_1^i, \cdots, \mathfrak{a}_{l+1}^i, \mathfrak{c}_1, \cdots, \mathfrak{c}_{(k-1)(l+1)}$.
4: Let $(H_k, (\mathfrak{c}_i)_{i\le k(l+1)}$ be the Hermite normal form of $(B_k, (\mathfrak{d}_i)_{i\le k(l+1)})$ modulo $\mathfrak{g}$ (by using [3, Alg. 1.6.1]).
5: **end for**
6: Let $H$ be the upper left $(l+1)\times(l+1)$ minor of $H_n$.
7: **return** $(H, (\mathfrak{c}_i)_{i\le l+1})$.

**Proposition 1.** *Let $M_1, \cdots, M_n$ be $\mathcal{O}_K$-modules of $K^{l+1}$ given by*

$$(A_1, (\mathfrak{a}_i^{(1)})_{i\le l+1}), \cdots, (A_n, (\mathfrak{a}_i^{(n)})_{i\le l+1}),$$

*where the $\mathfrak{a}_i^j$ are integral ideals of $\mathcal{O}_K$ and the $A_i$ have integral coefficients, then Algorithm 2 computes a pseudo-basis for $\cap_i M_i$ in polynomial time in $n$, $l$ and in the bit size of $\mathfrak{g} := \mathfrak{g}(M_1)\cdots\mathfrak{g}(M_n)$.*

*Proof.* We notice that $\forall k \le n$, $\mathfrak{g}(H_k) = \prod_{i=1}^k \mathfrak{g}(M_i)$ since it is clear that in the square case, the Hemite normal form computation preserves the determinantal ideal. Therefore, since the $\mathfrak{a}_i^j$ are assumed to be integral and since the $A_i$ have integral coefficients, $\mathfrak{g}$ is an integral multiple of $\mathfrak{g}(H_k)$ for every $k \le n$. Thus, the use of [3, Alg. 1.6.1] modulo $\mathfrak{g}$ in Step 4 is valid, and from [6, Th. 1] we know that it is polynomial in $n, l$ and the bit size of $\mathfrak{g}$.

We can see by induction that for all $k \leq n$, the pseudo-matrix defined by the upper left $(l+1) \times (l+1)$ minor of $H_k$ and $\mathfrak{c}_1, \cdots, \mathfrak{c}_{l+1}$ represent $\cap_{i \leq k} M_i$. The first iteration over $k$ is a direct application of [3, Alg. 1.5.1]. Let $M, N$ are two modules over a Dedekind domain given by $A, (\mathfrak{a}_i)_{i \leq l+1}$ and $B, (\mathfrak{b}_i)_{i \leq l+1}$, and let $C$ be the pseudo-matrix given by

$$\begin{pmatrix} A & A \\ 0 & B \end{pmatrix},$$

and $\mathfrak{c}_1, \cdots, \mathfrak{c}_{2(l+1)}$. Then the upper left $(l+1) \times (l+1)$ minor for the HNF of $C$, together with $\mathfrak{c}_1, \cdots, \mathfrak{c}_n$ represent $M \cap N$. Now if we assume our property for some $k < n$, then $B_{k+1}$ has the shape

$$B_{k+1} = \left(\begin{array}{cc:c} A_{k+1} & A_{k+1} & \\ (0) & H'_1 & (0) \\ \hdashline H'_3 & & H'_2 \end{array}\right),$$

where $H'_2 \in K^{(k-1)(l+1) \times (k-1)(l+1)}$ is lower-triangular and $(H'_1, (\mathfrak{c}_i)_{i \leq l+1})$ represent $\cap_{i \leq k} M_i$ (by induction).

The HNF of $B_{k+1}$ is obtained by using [3, Alg. 1.6.1]. The treatment of columns $(k+1)(l+1)$ to $2(l+1)+1$ leaves $H'_2$ and $\mathfrak{c}_{(k+1)(l+1)}, \cdots, \mathfrak{c}_{2(l+1)+1}$ unchanged since they are already in HNF. Then the treatment of columns $2(l+1)$ to 1 consists of computing the HNF of the pseudo-matrix

$$C_{k+1} := \begin{pmatrix} A_{k+1} & A_{k+1} \\ 0 & H'_1 \end{pmatrix}, (\mathfrak{a}_1^{(k+1)}, \cdots, \mathfrak{a}_{l+1}^{(k+1)}, \mathfrak{c}_1, \cdots, \mathfrak{c}_{l+1}),$$

and of reducing the rows of $H'_3$ with these $C_{k+1}$. Let $\left(UC_{k+1}, (\mathfrak{d}_i)_{i \leq 2(l+1)}\right)$ be the HNF of $(C_{k+1}, I)$, where $U \in K^{2(l+1) \times 2(l+1)}$, and $H'_4$ be the matrix defined by

$$U \begin{pmatrix} A_{k+1} & A_{k+1} \\ 0 & H'_1 \end{pmatrix} = \begin{pmatrix} H'_4 & (0) \\ (*) & (*) \end{pmatrix},$$

then by induction, we know that the pseudo-matrix

$$(H'_4, (\mathfrak{d}_i)_{i \leq l+1})$$

represents $\cap_{i \leq k+1} M_i$.

**Remark:** In the previous proof, the HNF of the upper left $2(l+1) \times 2(l+1)$ minor of $B_k$ has to be recovered from the HNF of $B_k$ itself since we need to perform this operation modulo an integral multiple of the determinantal ideal. We know such a multiple for $B_k$, but not for its upper left $2(l+1) \times 2(l+1)$ minor whose determinantal ideal might be fractional.

**Algorithm 3** Computation of the determinantal ideal

**Input:** Pseudo-matrix $A, (\mathfrak{a}_i)_{i\leq l+1}$, where $A$ and the $\mathfrak{a}_i$ are integral. Bound $L \geq \max_{i,j,\sigma} |\sigma(A_{i,j})|$.

**Output:** $\mathfrak{g}(A, (\mathfrak{a}_i))$.

1: $L_p \leftarrow \sqrt{d} L^{l+1} (l+1)^{(l+1)/2}$.
2: Let $p \in \mathbb{N}$ be the smallest prime above $L_p$ and $(\mathfrak{p}_i, e_i)_{i\leq g}$ such that $(p) = \prod_{i\leq g} \mathfrak{p}_i^{e_i}$.
3: **for** $\mathfrak{p}_i \mid (p)$ **do**
4: Compute $\det(A) \bmod \mathfrak{p}_i$.
5: **end for**
6: Recover $\det(A) \bmod (p)$ via

$$\begin{array}{rcl} \mathcal{O}_K/\mathfrak{p}_1^{e_1} \times \cdots \times \mathcal{O}_K/\mathfrak{p}_g^{e_g} & \longrightarrow & \mathcal{O}_K/(p) \\ (\det(A) \bmod \mathfrak{p}_1^{e_1}, \cdots, \det(A) \bmod \mathfrak{p}_g^{e_g}) & \longmapsto & \det(A) \bmod (p). \end{array}$$

7: $\mathfrak{g} \leftarrow \det(A)\mathfrak{a}_1 \cdots \mathfrak{a}_{l+1}$.
8: **return** $\mathfrak{g}$.

Before presenting the overall strategy for computing our decoding polynomial $c$, let us isolate the step consisting of the computation of the determinantal ideal of $J_i^{z_i}$. Algorithm 3 computes the determinantal ideal in polynomial time in $d$, $l$ and the bit size of $L$. In Step 1 we first compute an upper bound on $|\sigma(\det(A))|$ for the $d$ complex embeddings of $K$ via Hadamard's inequality and then we deduce a bound $L_p$ on $\|\det(A)\|$. If $\alpha_1, \cdots, \alpha_d$ is an integral basis for $\mathcal{O}_K$, and $(a_i)_{i\leq d}$ are integers such that $\det(A) = \sum_{i\leq d} a_i\alpha_i$, then necessarily $\max_i |a_i| \leq L_p$. Then, the unique representant of the class of $\det(A) \bmod (p)$ whose integral coefficients have their absolute value bounded by $p$ is $\det(A)$. Note that for the purpose of a practical implementation, one might prefer computing $\det(A)$ modulo several wordsize primes and recover its value via the chinese remaindering theorem rather than finding directly its value modulo a unique prime larger than $L_p$. Let us now present the overall strategy for computing the decoding polynomial.

**Proposition 2.** *Algorithm 4 is correct and runs in polynomial time in* $\log(N)$, $l$, $d$, *and* $\log|\Delta|$.

*Proof.* The loop between Step 1 and Step 8 of Algorithm 4 consists of computing a pseudo-basis for $J_i^{z_i}$ along with its determinantal ideal. These steps are polynomial in $K$, $l$, $d$ and $\log|\Delta|$. All we need to prove is that $L$ is indeed an upper bound on $\max_{k,g,\sigma} |\sigma(A_{i\,k,g})|$ where $\sigma : K \to \mathbb{C}$ runs over the $d$ complex embeddings of $K$. Every $r_k$ can be written $r_k = \sum_{g\leq d} b_g^{(k)}\alpha_g$ where $(\alpha_g)_{g\leq d}$ is an integral basis of $\mathcal{O}_K$. In addition, according to [6, Lem. 1], we may assume that $\|\alpha_g\| \leq \sqrt{d} 2^{d^2/2} \sqrt{|\Delta|}$. Trivially, we see that our canonical choice for a representative of a class modulo $\mathfrak{p}_k$ satisfies $|b_g^{(k)}| \leq \mathcal{N}(\mathfrak{p}_k)$. We thus have

$$\forall \sigma, k,\ |\sigma(r_k)| \leq \|r_k\| \leq \mathcal{N}(\mathfrak{p}_k)\sqrt{d} 2^{d^2/2} \sqrt{|\Delta|} \leq N\sqrt{d} 2^{d^2/2} \sqrt{|\Delta|}.$$

**Algorithm 4** Computation of the decoding polynomial

**Input:** $(\mathfrak{p}_i, z_i)_{i\leq n}$, $l$, $N$, $B$, $F$ such that $\exists c \in \prod_i J_i^{z_i}$ of degree at most $l$ satisfying (2) for $F$, and the encoded message $(r_1, \cdots, r_n) \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$.

**Output:** $c \in \prod_i J_i^{z_i}$ satisfying (2) for $F' = 2^{\frac{dl}{2}}\sqrt{l+1}\left(2^{2+d(6+3d)}d^3|\Delta|^{2+\frac{11}{2d}}\right)F$ of degree at most $l$.

1: $L \leftarrow 2^{ld^2/2}\binom{2l}{l}N^l|\Delta|^{l/2}$.
2: **for** $i \leq n$ **do**
3: $\quad \tilde{z_i} \leftarrow \min(z_i, l)$.
4: $\quad$ For $0 \leq j \leq \tilde{z_i}$: $\mathfrak{a}_j^i \leftarrow \mathfrak{p}_i^{z_i - j}$, $a_j^i \leftarrow (y - r_i)^j$.
5: $\quad$ For $1 \leq j \leq l - z_i$: $\mathfrak{a}_j^i \leftarrow \mathcal{O}_K$, $a_j^i \leftarrow y^j(y - r_i)^{z_i}$.
6: $\quad$ Let $\left(A_i, (\mathfrak{a}_j^i)_{j\leq l+1}\right)$ be the pseudo matrix representing $J_i^{z_i}$.
7: $\quad$ Compute $\mathfrak{g}(J_i^{z_i})$ with Algorithm 3 used with the bound $L$.
8: **end for**
9: Compute a pseudo-basis $[(c_i), (\mathfrak{c}_i)]_{i\leq l+1}$ of $M = \cap_i J_i^{z_i}$ modulo $\mathfrak{g} = \prod_i \mathfrak{g}(J_i^{z_i})$ with algorithm 2.
10: Deduce a pseudo basis $[(d_i), (\mathfrak{d}_i)]_{i\leq l+1}$ of the module $M'$ given by

$$(v_0, v_1, \cdots, v_l) \in M \iff (v_0, v_1 \cdot B, \cdots, v_l \cdot (B)^l) \in M'.$$

11: Let $[(b_i), (\mathfrak{b}_i)]_{i\leq l+1}$ be a short peudo-basis of $M'$ obtained with the reduction algorithm of [6].
12: Let $x_1, x_2$ be a short basis of $\mathfrak{b}_1$ obtained with [6, Th. 3].
13: **return** $c \in M$ corresponding to $x_1 b_1 \in M'$.

The coefficients of $A$ are given by those of polynomials of the form $(y - r_k)^g$ where $g \leq l$. They are bounded by

$$\max_{\sigma,k,g} |\sigma(A_{i\,k,g})| \leq \binom{2l}{l}(\max_{\sigma,k} |\sigma r_k|)^l \leq 2^{ld^2/2}\binom{2l}{l}N^l|\Delta|^{l/2}.$$

Givem the bounds on $\|\det(A_i)\|$, we clearly see that $\mathfrak{g} = \prod \mathfrak{g}(J_i^{z_i})$ has bit size polynomial in $\log(N)$, $l$, $d$ and $\log|\Delta|$, which implies that step 9 is polynomial in all these values.

By assumption, we know the existence of $e \in \prod_i J_i^{z_i}$ of degree at most $l$ satisfying (2) for $F$. We even argued in the proof of Lemma 2 that we could assume

$$\forall j \leq l,\ \|e_j\| \leq \frac{F}{(l+1)B^j}.$$

The size of the vector $e'$ corresponding to $e$ in the module $M'$ defined in Step 10 satisfies $\|e'\| \leq F/\sqrt{l+1}$, and so does the first minima $\lambda_1(M')$, that is $\lambda_1(M') \leq \|c'\| \leq F/\sqrt{l+1}$. Note that we use $\|e\| = \sqrt{\sum_i \|e\|^2}$ to define the size of an element of an $\mathcal{O}_K$-module. From [6, Th. 3], we know that $\mathcal{N}(\mathfrak{b}_1) \in [2^{-O(d^2)}, 1]$ and that

$$\begin{aligned}\|x_1\| &\leq 4 \cdot 2^{4d}|\Delta|^{4/d}\left(\max_{i\leq d}\|\alpha_i\|\right)^4 \mathcal{N}(\mathfrak{b}_1)^{4/d} \\ &\leq 2^{2(1+d(2+d))}d^2|\Delta|^{2\left(1+\frac{2}{d}\right)}.\end{aligned}$$

In addition, $\|b_1\|$ can be bounded by using [6, Cor. 3]

$$\begin{aligned}\|b_1\| &\leq 2^{\frac{d(l+1)}{2}}\sqrt{l+1}2^{\frac{3d}{2}}|\Delta|^{\frac{3}{2d}}\left(\max_{i\leq d}\|\alpha_i\|\right)^2\lambda_1(M') \\ &\leq 2^{d\left(\frac{l+1}{2}+\frac{3}{2}+d\right)}\sqrt{l+1}d|\Delta|^{\frac{3}{2d}}\lambda_1(M').\end{aligned}$$

We thus have that

$$\|x_1 b_1\| \leq \left(2^{2+d\left(\frac{l+12}{2}+3d\right)}d^3|\Delta|^{2+\frac{11}{2d}}\right)F,$$

and therefore, for every $m \in \mathcal{O}_K$ such that $\|m\| \leq B$, the polynomial $c \in M$ corresponding to $x_1 b_1 \in M'$ satisfies

$$\|c(m)\| \leq \sqrt{l+1}\|x_1 b_1\| \leq \sqrt{l+1}\|x_1\|\|b_1\| \leq 2^{\frac{dl}{2}}\sqrt{l+1}\left(2^{2+d(6+3d)}d^3|\Delta|^{2+\frac{11}{2d}}\right)F.$$

## 8 Good weight settings

To derive our main result, we need to consider weights $z_i > 0$ in $\mathbb{R}$ rather than $\mathbb{Z}$. Let

$$\beta_{d,\Delta,\gamma} := \frac{d^{3-\frac{d}{2}}2^{3(1+d(2+d))}|\Delta|^{2+\frac{11}{2d}}}{\alpha_{d,\Delta,\gamma}{}^{\frac{1}{d}}},$$

then by combining (3), (4) and Algorithm 4, we know that given $(r_1, \cdots, r_n) \in \prod_{i\leq n}\mathcal{O}_K/\mathfrak{p}_i$, $l > 0$, $B = \prod_{i\leq k}\mathcal{N}(\mathfrak{p}_i)$ and integer weights $z_i > 0$, Algorithm 4 returns a polynomial $c$ of degree at most $l$ such that all $m \in \mathcal{O}_K$ satisfying $\|m\| \leq B$ and

$$\begin{aligned}\sum_{i\leq n} a_i z_i \log\mathcal{N}(\mathfrak{p}_i) \geq &\frac{l}{2}\log(2^{d^2}B^d) + \frac{3d}{2}\log(l+1) \\ &+ \frac{1}{l+1}\sum_{i\leq n}\binom{z_i+1}{2}\log\mathcal{N}(\mathfrak{p}_i) + \log\beta_{d,\Delta,\gamma}, \qquad (5)\end{aligned}$$

(where $a_i = 1$ if $m \bmod \mathfrak{p}_i = r_i$, 0 otherwise) are roots of $c$. In the following, we no longer assume the $z_i$ to be integers. However, we will use our previous results with the integer weights $z_i^* := \lceil Az_i \rceil$ for a sufficently large integer $A$ to be determined.

**Proposition 3.** *Let $\varepsilon > 0$, non-negative reals $z_i$, $B = \prod_{i\leq k}\mathcal{N}(\mathfrak{p}_i)$, and an encoded message $(r_1, \cdots, r_n) \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$, then our algorithm finds all the $m \in \mathcal{O}_K$ such that $\|m\| \leq B$ and*

$$\sum_{i\leq n} a_i z_i \log\mathcal{N}(\mathfrak{p}_i) \geq \sqrt{\log(2^{d^2}B^d)\left(\sum_{i\leq n} z_i^2\log\mathcal{N}(\mathfrak{p}_i) + \varepsilon z_{max}^2\right)},$$

*where $a_i = 1$ if $m \bmod \mathfrak{p}_i = r_i$, 0 otherwise.*

*Proof.* Note that we can assume without loss of generality that $z_{max} = 1$. Let $z_i^* = \lceil Az_i \rceil$ for a sufficently large integer $A$, which thus satisfies $Az_i \leq z_i^* < Az_i + 1$. The decoding condition (5) is met whenever

$$\sum_{i \leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) \geq \frac{l}{2A} \log(2^{d^2} B^d) + \frac{3d}{2A} \log(l+1) + \frac{A}{2(l+1)} \sum_{i \leq n} \left( z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2} \right) \log \mathcal{N}(\mathfrak{p}_i) + \frac{1}{A} \log \beta_{d,\Delta,\gamma}. \tag{6}$$

Let $Z_i := z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2}$ for $i \leq n$ and

$$l := \left\lceil A \sqrt{\frac{\sum_{i \leq n} Z_i \log \mathcal{N}(\mathfrak{p}_i)}{\log(2^{d^2} B^d)}} \right\rceil - 1.$$

We assume that $A \geq \log(2^{d^2} B^d)$, which ensures that $l > 0$. For this choice of $l$, condition (6) is satisfied whenever

$$\sum_{i \leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) \geq \frac{3d}{2A} \log \left( A \sqrt{\frac{\sum_{i \leq n} Z_i \log \mathcal{N}(\mathfrak{p}_i)}{\log(2^{d^2} B^d)}} + 1 \right) + \sqrt{\log(2^{d^2} B^d) \left( \sum_{i \leq n} Z_i \log \mathcal{N}(\mathfrak{p}_i) \right)} + \frac{1}{A} \log \beta_{d,\Delta,\gamma}. \tag{7}$$

Assume that $A \geq \frac{10 \log N}{\varepsilon}$ and $A \geq \frac{\log \beta_{d,\Delta,\gamma}}{\log N}$, then for $N$ large enough, the right side of (7) is at most

$$O\left( \frac{\log \log N}{\log N} \right) + \sqrt{\log(2^{d^2} B^d) \left( \sum_{i \leq n} z_i^2 \log \mathcal{N}(\mathfrak{p}_i) + \frac{\varepsilon}{2} \right)}$$
$$\leq \sqrt{\log(2^{d^2} B^d) \left( \sum_{i \leq n} z_i^2 \log \mathcal{N}(\mathfrak{p}_i) + \varepsilon \right)}$$

The degree $l$ of our decoding polynomial $c$ is therefore polynomial in $\log N$, $\frac{1}{\varepsilon}$, $d$ and $\log |\Delta|$. By [1, 2.3], we know that the complexity to find the roots of $c$ is polynomial in $d$, $l$ and in the logarithm of the height of $c$, which we already proved to be polynomial in the desired values.

**Corollary 1.** *Let $\varepsilon > 0$, $k < n$ and prime ideals $\mathfrak{p}_1, \cdots \mathfrak{p}_n$ satisfying $\mathcal{N}(\mathfrak{p}_i) < \mathcal{N}(\mathfrak{p}_{i+1})$ and $\log \mathcal{N}(\mathfrak{p}_{k+1}) \geq \max(2dk \log \mathcal{N}(\mathfrak{p}_k), 2d^2)$, then with the previous notations, our algorithm finds a list of all codewords which agree with a received word in $t$ places provided $t \geq \sqrt{k(n+\varepsilon)}$.*

*Proof.* The proof is similar to the one of [10, Th. 7.14]. The main difference is that we define $\delta := k - \frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})}$ which satisfies $\delta \geq 0$ since by assumption $\log \mathcal{N}(\mathfrak{p}_{k+1}) \geq \max(2dk \log \mathcal{N}(\mathfrak{p}_k), 2d^2)$. We apply Proposition 3 with $z_i = 1/\log \mathcal{N}(\mathfrak{p}_i)$ for $i \geq k+1$, $z_i = 1/\log \mathcal{N}(\mathfrak{p}_{k+1})$ for $i \leq k$, and $\varepsilon' = \varepsilon/\log \mathcal{N}(\mathfrak{p}_{k+1})$. It allows us to retrieve the codewords whose number of agreements $t$ is at least

$$\sqrt{\frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})}\left(\frac{\log(B)}{\log \mathcal{N}(\mathfrak{p}_{k+1})} + \sum_{i=k+1}^{n} \frac{\mathcal{N}(\mathfrak{p}_{k+1})}{\log \mathcal{N}(\mathfrak{p}_i)} + \varepsilon'\right)}$$
$$\leq \delta + \sqrt{\frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})}\left(\frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})} + \sum_{i=k+1}^{n} \frac{\mathcal{N}(\mathfrak{p}_{k+1})}{\log \mathcal{N}(\mathfrak{p}_i)} + \varepsilon\right)}.$$

This condition is met whenever $t \geq \delta + \sqrt{(k-\delta)(n-\delta+\varepsilon)}$. From the Cauchy-Schwartz inequality, we notice that

$$\sqrt{k(n+\varepsilon)} \geq \sqrt{(k-\delta)(n-\delta+\varepsilon)},$$

which proves that our decoding algorithm works when $t \geq \sqrt{k(n+\varepsilon)}$.

## 9 Conclusion

We derived an analogue of the CRT list decoding algorithm for codes based on number fields. Our method allows to reach the Johnson bound. We followed the approach of [10, Ch. 7] that provides a general frameworks for list decoding of algebraic codes, along with its application to CRT codes. The modifications to make this strategy efficient in the context of number fields are substantial. We needed to refer to the theory of modules over a Dedekind domain, and carefully analyse the process of intersecting them, as well as finding a short basis. Our algorithm is polynomial in $d$, $\log(N)$ and in $\log|\Delta|$. The only regret we can have is that we need to assume $\log \mathcal{N}(\mathfrak{p}_{k+1}) \geq \max(2dk \log \mathcal{N}(\mathfrak{p}_k), 2d^2)$, which is the only difference with the result of [10, Ch. 7] on CRT codes.

We could not conclude on whether the numer field codes can allow better decoding settings than CRT codes. So far, we have shown that they were effective and at least as good as CRT codes. Other types of decoding could be tried. In addition, more fundamental questions could be raised, for example on the performances of the dual code of a number field code.

## References


1. A. Ayad, *A lecture on the complexity of factoring polynomials over global fields*, International Mathematical Forum **5** (2010), no. 10, 477–486.
2. D. Boneh, *Finding smooth integers in short intervals using crt decoding*, Proceedings of the thirty-second annual ACM symposium on Theory of computing (New York, NY, USA), STOC '00, ACM, 2000, pp. 265–272.

3. H. Cohen, *Advanced topics in computational algebraic number theory*, Graduate Texts in Mathematics, vol. 193, Springer-Verlag, 1991.
4. H. Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics, vol. 138, Springer-Verlag, 1991.
5. H. Cohn and N. Heninger, *Ideal forms of coppersmith's theorem and guruswami-sudan list decoding*, Proceedings of Innovations in computer science, 2011.
6. C. Fieker and D. Stehlé, *Short bases of lattices over number fields*, Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010. Proceedings (G. Hanrot, F. Morain, and E. Thomé, eds.), Lecture Notes in Computer Science, vol. 6197, Springer, 2010, pp. 157–173.
7. A. Garcia and H. Stichtenoth, *A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-vlăduţ bound*, Invent. Math. **121** (1995), no. 1, 211–222.
8. O. Goldreich, D. Ron, and M. Sudan, *Chinese remaindering with errors*, Proceedings of the thirty-first annual ACM symposium on Theory of computing (New York, NY, USA), STOC '99, ACM, 1999, pp. 225–234.
9. V. Guruswami, *Constructions of codes from number fields*, IEEE Transactions on Information Theory **49** (2003), no. 3, 594–603.
10. V. Guruswami, *List decoding of error-correcting codes: Winning thesis of the 2002 acm doctoral dissertation competition (lecture notes in computer science)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
11. V. Guruswami, A. Sahai, and M. Sudan, *Soft-decision decoding of chinese remainder codes*, Proceedings of the 41st Annual Symposium on Foundations of Computer Science (Washington, DC, USA), IEEE Computer Society, 2000, pp. 159–168.
12. V. Guruswami and M. Sudan, *Improved decoding of reed-solomon and algebraic-geometric codes*, IEEE Symposium on Foundations of Computer Science, vol. 5, 1999, pp. 28–39.
13. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), no. 4, 515–534.
14. H. Lenstra, *Codes from algebraic number fields*, Mathematics and computer science II, Fundamental contributions in the Netherlands since 1945 (North-Holland, Amsterdam) (M. Hazewinkel, J.K. Lenstra, and L.G. L.T. Meertens, eds.), CWI Monograph, vol. 4, 1986, pp. 94–104.
15. D. Mandelbaum, *On a class of arithmetic codes and a decoding algorithm (corresp.)*, IEEE Transactions on Information Theory **22** (1976), 85–88.
16. J. Neukirch, *Algebraic number theory*, Comprehensive Studies in Mathematics, Springer-Verlag, 1999, ISBN 3-540-65399-6.
17. M. A. Shokrollahi and H. Wasserman, *List decoding of algebraic-geometric codes*, IEEE Transactions on Information Theory **45** (1999), no. 2, 432–437.
18. M. A. Tsfasman and Th. Zink, *Modular curves, shimura curves, and goppa codes, better than varshamov-gilbert bound*, Mathematische Nachrichten **109** (1982), 21–28.