Gulf Journal of Mathematics
Vol 4, Issue 4 (2016) 189-205

PADÉ APPROXIMANTS AND KEY LATTICES FOR DECODING ALTERNANT CODES

K. ABDELMOUMEN[1*], H. BEN-AZZA[2], A. OTMANI[3]

Abstract. In this paper we present two methods for algebraic decoding of alternant codes. The paper recalls fundamental facts of these codes and the concept of key equation introduced by Berlekamp on which decoding is based. The first method is based on Padé approximants. The second method consist of translating the key equation into a lattice over the ring of polynomials, called *key lattice* which depends on the syndrome, and then seek a shortest vector in this key lattice, with degrees constraints. The key lattice approach, following the fruitfull formulation of H. W. Lenstra, is based on a result of Grigor'ev showing that computing a minimal vector in the mentioned lattice is polynomial. The computational complexity of the proposed methods are given in terms of the error-correction capability of the codes and results in tractable polynomial time algorithms. A fast parallel version of the key lattice method is deduced.

1. Introduction

We will study a "hard" decoding method based on a lattice over the ring of polynomials. Here, by a lattice we mean a finitely generated free module over a polynomial ring, adopting the formulation of [13]. Other formulations are possible as [14, 10, 9]. We present a method for algebraic decoding of alternant codes which consist of translating the key equation ([19, 6]) into a lattice over the ring of polynomials, called *key lattice* which depends on the syndrome, and seek the shortest vector in this key lattice. Thus we reduce this problem to a polynomially feasible optimization problem, thanks to the work of D. Yu. Grigor'ev [8, theorem 1.1 p. 1766] or that of von zur Gathen [10] highlighting valuation rings theory.

Furthermore, we will briefly review the decoding problem with Padé approximation (cf. [7, paragraph 8], [16, 2]), which can be solved by the techniques of continued fractions or the extended Euclidean algorithm.

For the convenience of the reader, we have organized this paper by indicating where to find known results and making it self-contained as much as possible. Section 2 contains the definitions and properties of alternant codes, for more information see [9, 15, 22]. Section 3 recalls the problem of decoding by key equation, the concept of rational reconstruction and Padé approximation (cf. [2])

Date: Accepted: Oct 24, 2016.
* Corresponding author.
2010 *Mathematics Subject Classification.* 68Q17, 68T30, 94A24.
Key words and phrases. Alternant codes, lattice, Padé approximants, key equation, decoding.

for decoding alternant codes. Section 4 contains the necessary and minimum definitions on lattices (cf. [13]) and a characterization theorem of the decoding by the novel concept of key lattice . Section 5 presents an algorithm for decoding. Furthermore, we give an estimate of its computational complexity which is refined to $\mathcal{O}(t \log^3 t)$, where t is the error-correction capability. A fast parallel version of the key lattice approach is deduced in subsection 5.1. In Section 6, we conclude the paper with some research perspectives.

2. Alternant Codes

A q-ary (linear) code $\mathcal{C}$ over the finite field $\mathbb{F}_q$ of q elements defined by a $k \times n$ matrix $\mathbf{G}$ (with $k \leq n$) whose entries belong to $\mathbb{F}_q$ is the vector space spanned by its rows *i.e.*,

$$\mathcal{C} \stackrel{\text{def}}{=} \Big\{ uG \mid u \in \mathbb{F}_q^k \Big\}.$$

The length of $\mathcal{C}$ is n and its rate is the ratio $R \stackrel{\text{def}}{=} k/n$.

One famous family of codes is the one of binary Goppa codes. It belongs to the more general class of alternant codes ([15, Chap. 12, p. 365]). The main well-known feature of an alternant code is the possibility of being decoded in polynomial time. It is more convenient to describe this class through a parity-check matrix over an extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ over which the code is defined. We recall that a parity-check matrix H of a q-ary code $\mathcal{C}$ is defined as a matrix such that:

$$\mathcal{C} \stackrel{\text{def}}{=} \Big\{ c \in \mathbb{F}_q^n \mid Hc^T = \mathbf{O} \Big\}.$$

Alternant codes are closely related to the generalized Reed-Solomon codes.

Definition 2.1 ([15])**.** Let $\alpha = (\alpha_1, ..., \alpha_n)$ where the α_i are distinct elements of $\mathbb{F}_{q^m}$, and let $v = (v_1, ..., v_n)$ where the v_i are nonzero (but not necessarily distinct) elements of $\mathbb{F}_{q^m}$.

$GRS_{k_0}(\alpha, v)$ consists of all vectors $(v_1 P(\alpha_1), ..., v_n P(\alpha_n))$ where $P(z)$ is any polynomial of degree $< k_0$ with coefficients from $\mathbb{F}_{q^m}$.

Theorem 2.2 ([15])**.** *$GRS_{k_0}(\alpha, v)$ is an $[n, k_0, r+1]$-MDS code over $\mathbb{F}_{q^m}$, where $r = n - k_0$, and has parity check matrix:*

$$\begin{aligned} H &= \begin{pmatrix} y_1 & y_2 & \cdots & y_n \\ \alpha_1 y_1 & \alpha_2 y_2 & \cdots & \alpha_n y_n \\ \cdots & \cdots & \cdots & \cdots \\ {\alpha_1}^{r-1} y_1 & {\alpha_2}^{r-1} y_2 & \cdots & {\alpha_n}^{r-1} y_n \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \cdots & \cdots & \cdots & \cdots \\ {\alpha_1}^{r-1} & {\alpha_2}^{r-1} & \cdots & {\alpha_n}^{r-1} \end{pmatrix} \begin{pmatrix} y_1 & & 0 \\ & \ddots & \\ 0 & & y_n \end{pmatrix} = XY \end{aligned} \tag{2.1}$$

where $y = (y_1, ..., y_n)$ (with $y_i \in \mathbb{F}_{q^m} \backslash \{0\}$) is such that $GRS_{k_0}(\alpha, v)^{\perp} = GRS_{n-k_0}(\alpha, y)$

Proposition 2.3 ([15]). *Let C be any invertible $r \times r$ matrix over $\mathbb{F}_{q^m}$. Then CH and H define the same code.*

Let $C = (c_{ij})$ be any invertible $r \times r$ matrix over $\mathbb{F}_{q^m}$. Also, a parity check matrix for $GRS_{k_0}(\alpha, v)$ is [15]:

$$H' = CXY = \begin{pmatrix} y_1 g_1(\alpha_1) & y_2 g_1(\alpha_2) & \cdots & y_n g_1(\alpha_n) \\ y_1 g_2(\alpha_1) & y_2 g_2(\alpha_2) & \cdots & y_n g_2(\alpha_n) \\ \cdots & \cdots & \cdots & \cdots \\ y_1 g_r(\alpha_1) & y_2 g_r(\alpha_2) & \cdots & y_n g_r(\alpha_n) \end{pmatrix} \tag{2.2}$$

where $g_i(x) = \sum_{j=1}^{r} c_{ij} x^{j-1}, 1 \leq i \leq r$, is a polynomial of degree $< r$ over $\mathbb{F}_{q^m}$.
Keeping the previous notations, we introduce:

Definition 2.4 ([15]). The alternant code $\mathcal{A}(\alpha, y, r)$ consists of all codewords of $GRS_{k_0}(\alpha, v)$ which have components from $\mathbb{F}_q$, i.e $\mathcal{A}(\alpha, y, r)$ is the restriction of $GRS_{k_0}(\alpha, v)$ to $\mathbb{F}_q$.

Thus $\mathcal{A}(\alpha, y, r)$ consists of all vectors c over $\mathbb{F}_q$ such that $Hc^{\intercal} = 0$, where H is given by Eq. 2.1 or Eq. 2.2.

Proposition 2.5 ([15]). *If $\mathcal{C} = \mathcal{A}(\alpha, y, r)$, then $n - r \geq dim(\mathcal{C}) \geq n - rm$ and $d_{\mathcal{C}} \geq r + 1$.*

Classical Goppa codes. Let g be a polynomial of degree $r > 0$ with coefficients from $\mathbb{F}_{q^m}$, and $L = \{\alpha_1, ..., \alpha_n\}$ where the α_i are distinct elements of $\mathbb{F}_{q^m}$ such that $g(\alpha_i) \neq 0$ for all i.

The classical Goppa code $\Gamma(L, g)$ = alternant code $\mathcal{A}(\alpha, y, r)$ where $y_i = g(\alpha_i)^{-1}$, $\alpha = (\alpha_1, ..., \alpha_n)$, and has a parity check matrix:

$$H = XY = \begin{pmatrix} g(\alpha_1)^{-1} & g(\alpha_2)^{-1} & \cdots & g(\alpha_n)^{-1} \\ \alpha_1 g(\alpha_1)^{-1} & \alpha_2 g(\alpha_2)^{-1} & \cdots & \alpha_n g(\alpha_n)^{-1} \\ \cdots & \cdots & \cdots & \cdots \\ {\alpha_1}^{r-1} g(\alpha_1)^{-1} & {\alpha_2}^{r-1} g(\alpha_2)^{-1} & \cdots & {\alpha_n}^{r-1} g(\alpha_n)^{-1} \end{pmatrix}$$

BCH Codes.

Proposition 2.6 ([9, 20]). *The strict BCH codes are classical Goppa codes.*

Let ω be a primitive element of $\mathbb{F}_{q^m}$ and let δ be an integer such that $2 \leq \delta \leq n$. The strict BCH code of length n and designed distance δ , coincides with classical Goppa code $\Gamma(\alpha, X^{\delta-1})$ with $\alpha = \left(1, \omega^{-1}, ..., \omega^{-(n-1)}\right)$.

3. Decoding Issues

Preliminary facts. We consider an alternant code $\mathcal{A}(\alpha, y, r)$ of degree r and length $n \leq q^m$. A decoding algorithm corrects r errors if it recovers u from the knowledge of $y \stackrel{\text{def}}{=} uG + e$ for all possible $e \in \mathbb{F}_q^n$ of weight $< r$ where G is a generator matrix. An equivalent way of describing the decoding is to consider the syndrome vector $S \stackrel{\text{def}}{=} Hy^T$ and finding e of minimum weight such that $S = He^T$. Let $I \stackrel{\text{def}}{=} \{i : e_i \neq 0\}$ be the set of error positions. For all $i \in I$, we call α_i an

error locator. Since the α_j are distinct, the knowledge of the error locators is equivalent to the knowledge of the error positions. We also define the error locator polynomial $\varphi(x)$ and the error evaluator polynomial $\psi(x)$ as:

$$\varphi(x) \stackrel{\text{def}}{=} \prod_{i\in I}(1-\alpha_i x) \quad \text{and} \quad \psi(x) \stackrel{\text{def}}{=} \sum_{i\in I} e_i y_i \prod_{j\in I: j\neq i}(1-\alpha_j x). \tag{3.1}$$

The roots of $\varphi(x)$ are precisely the reciprocals of the error locators. Errors are evaluated by the following Forney's formula:

$$\forall i \in I,\ e_i = -\frac{\alpha_i}{y_i}\frac{\psi}{\varphi'}\left(\alpha_i^{-1}\right) \tag{3.2}$$

where φ' is the derivative of φ. The relation (3.2) shows that the knowledge of $\psi(x)$ and $\varphi(x)$ permits to recover the error vector e. Actually these two quantities are closely related to the polynomial syndrome $S(x) \stackrel{\text{def}}{=} \sum_{u=0}^{r-1} s_u x^u$. It is clear that:

$$\begin{aligned}
S(x) &= \sum_{u=0}^{r-1} s_u x^u \\
&= \sum_{u=0}^{r-1}\left(\sum_{i\in I} e_i y_i \alpha_i^u\right) x^u \\
&= \sum_{i\in I} e_i y_i \sum_{u=0}^{r-1}(x\alpha_i)^u \\
&= \sum_{i\in I} e_i y_i \prod_{j\in I: j\neq i}(1-\alpha_j x)\frac{(1-\alpha_i x)}{\varphi(x)}\sum_{u=0}^{r-1}(x\alpha_i)^u.
\end{aligned}$$

This implies that:

$$\begin{aligned}
\varphi(x)S(x) &= \sum_{i\in I} e_i y_i \prod_{j\in I: j\neq i}(1-\alpha_j x)\left(1-(x\alpha_i)^r\right) \\
&= \psi(x) - x^r\left(\sum_{i\in I} e_i y_i \alpha_i^r \prod_{j\in I: j\neq i}(1-\alpha_j x)\right) \\
&= \psi(x) \pmod{x^r}.
\end{aligned}$$

The key equation defined by (3.3) below admits at least one solution if the number of unknowns $\deg(\varphi)+\deg(\psi)$ is smaller than or equal to r, that is to say, $2\deg(\varphi) \leq r$. The degree of φ corresponds to number of errors that corrupted the codewords. Usually we denote by t such a number and therefore we will assume from now that $\deg(\varphi) \leq t$ where we have set $t \stackrel{\text{def}}{=} \lfloor r/2 \rfloor$. We have hence proved the following proposition.

Proposition 3.1. *The polynomials $\varphi(x)$, $S(x)$ and $\psi(x)$ satisfy the key equation:*

$$\varphi(x)S(x) = \psi(x) \pmod{x^r}. \tag{3.3}$$

with $\varphi(0)=1$ and $\deg(\psi) < \deg(\varphi) \leq t$ where $t = \lfloor r/2 \rfloor$.

Lemma 3.2 ([5]). *Let ϕ_1 and ψ_1 be two polynomials over $\mathbb{F}_q$ satisfying the key equation (3.3) with $deg(\phi_1) \leq t$ and $deg(\psi_1) < t$. There exists a polynomial $\lambda \in \mathbb{F}_q[x]$ with $\phi_1(x) = \lambda(x)\varphi(x)$ and $\psi_1(x) = \lambda(x)\psi(x)$*

Solving the key equation. The decoding of alternant codes can be seen as a special case of the following well-known problem called rational reconstruction [11].

Definition 3.3 ([11]). Let $\mathbb{K}$ be a field, g in $\mathbb{K}[x]$ of degree $r > 0$ and f in $\mathbb{K}[x]$ of degree $< r$. For a given h such that $1 \leq h \leq r$, the rational reconstruction of f modulo g consists in finding $(\psi, \varphi) \in \mathbb{K}[x]^2$ such that $\gcd(\varphi, g) = 1$, $\deg(\psi) < h$ and $deg(\varphi) \leq r - h$ verifying:

$$\frac{\psi}{\varphi} = f \pmod{g} \tag{RR}$$

When g is the polynomial x^r then we rather speak of Padé approximation.

Note that if (RR) has a solution (ψ, φ) then $\frac{\psi}{\varphi}$ is unique. Furthermore if $(\psi, \varphi) \in \mathbb{K}[x]^2$ is a solution of the problem (RR), then (ψ, φ) also verifies the following problem.

Definition 3.4. Let $\mathbb{K}$ be a field, $g \in \mathbb{K}[x]$ a polynomial of degree $r > 0$, $f \in \mathbb{K}[x]$ of degree $< r$. For a given h such that $1 \leq h \leq r$, we want to find ψ, φ in $\mathbb{K}[x]$ such that $(\varphi, \psi) \neq (0, 0)$ and:

$$\varphi f = \psi \pmod{g} \text{ with } \deg(\psi) < h \text{ and } \deg(\varphi) \leq r - h. \tag{RRS}$$

Clearly, any solution to (RRS) is solution to (RR) if and only if $\gcd(\varphi, g) = 1$. Moreover, (RRS) always has a non-trivial solution since recovering φ and ψ can be done by solving a linear system with r equations satisfying by $h+(r-h+1) = r+1$ unknowns representing the coefficients of φ and ψ.

A classical way of solving (RR) is to apply the extended euclidean algorithm (EEA) to (f, g). Recall that if we denote by (U_i, V_i, R_i), with $i \geq 0$, the polynomials obtained at the i-th step of $\mathsf{EEA}(f, g)$ then we have $R_0 \stackrel{\text{def}}{=} g$, $R_1 \stackrel{\text{def}}{=} f$ and for all $i \geq 0$:

$$\begin{cases} R_i = U_i f + V_i g \\ \text{ with } (U_0, U_1) \stackrel{\text{def}}{=} (0, 1) \text{ and } (V_0, V_1) \stackrel{\text{def}}{=} (1, 0) \\ R_i = Q_{i+1} R_{i+1} + R_{i+2} \\ \text{ with } 0 \leq \deg(R_{i+2}) < \deg(R_{i+1}) \end{cases}.$$

We also have the relations $U_{i+2} = -Q_{i+1}U_{i+1} + U_i$ and $V_{i+2} = -Q_{i+1}V_{i+1} + V_i$. We are now able to prove that this approach will provide a non-trivial solution.

Proposition 3.5 ([11][Chapter 5]). *Let j be the smallest integer such that $\deg(R_j) < h$ then (R_j, U_j) is a non-trivial solution to (RRS).*

Proof. One important fact that will be useful in the proof is that at each step $i \geq 0$ of $\mathsf{EEA}(f, g)$ it holds that:

$$\deg(U_{i+1}) = r - \deg(R_i). \tag{3.4}$$

This can be proven by induction on i. This is true when $i = 0$ since $U_1 = 1$ and $R_0 = g$. Assume now that (3.4) is true at rank i, that is to say, $\deg(U_{i+1}) = r - \deg(R_i)$. By definition of R_i, we have $\deg(R_i) = \deg(Q_{i+1}) + \deg(R_{i+1})$. Observe also that $\deg(U_{i+2}) = \max\{\deg(Q_{i+1}) + \deg(U_{i+1}), \deg(U_i)\}$. But, it is easy to see $\deg(U_i) \leq \deg(U_{i+1})$. So it turns out that:

$$\begin{aligned} \deg(U_{i+2}) &= \deg(Q_{i+1}) + \deg(U_{i+1}) \\ &= (\deg(R_i) - \deg(R_{i+1})) + (r - \deg(R_i)) \\ &= r - \deg(R_{i+1}). \end{aligned}$$

We now have to prove that $\deg(U_j) \leq r - h$. But by definition of j we necessarily have $\deg(R_{j-1}) \geq h$ and hence $\deg(U_j) \leq r - h$ thanks to (3.4).

□

Corollary 3.6 ([11]). *The cost for solving* (RRS) *in the worst case is* $\mathcal{O}(r^2)$.

The following proposition shows that (R_j, U_j) provides essentially a unique solution to (RR) when it exists.

Proposition 3.7 ([11]). *If* (φ, ψ) *is a solution to* (RR) *then there exists* λ *in* $\mathbb{K}\backslash\{0\}$ *such that* $\varphi = \lambda U_j$ *and* $\psi = \lambda R_j$.

We have all the material to devise a decoding procedure in the case of alternant codes. If $S(x) \neq 0$ then we apply $\mathsf{EEA}(S, x^r)$ in order to obtain the smallest j such that $\deg(R_j) < t$. If such j exists then choose λ in $\mathbb{F}_{q^m}\backslash\{0\}$ such that $\lambda U_j(0) = 1$. Next we set $\varphi = \lambda U_j$ and $\psi = \lambda R_j$. The last step of the decoding consists then in finding the roots of φ in order to locate the error positions and finally the error magnitudes thanks to (3.2). If we assume that we perform a basic search in the field $\mathbb{F}_{q^m}$ then the cost of root finding is $\mathcal{O}(tq^m)$ and hence the overall complexity of decoding is $\mathcal{O}(t^2 + tq^m)$.

Alternative approaches. Another possible way of solving (3.3) is to set up a linear system where the unknowns are the coefficients of the polynomials $\varphi(x)$ and $\psi(x)$, say $\varphi(x) \stackrel{\text{def}}{=} \sum_{u=0}^{t} \phi_u x^u$ and $\psi(x) \stackrel{\text{def}}{=} \sum_{u=0}^{t-1} \psi_u x^u$. Recall that by assumption $\phi_0 = 1$ and $\psi_0 = s_0$. For the sake of simplicity we assume that $r = 2t$. From the key equation we deduce the following two linear systems defined by Henkel matrices:

$$\begin{pmatrix} s_0 & 0 & \cdots & 0 \\ s_1 & s_0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ s_{t-1} & s_{t-2} & \cdots & s_0 \end{pmatrix} \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{t-1} \end{pmatrix} = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{t-1} \end{pmatrix} \tag{3.5}$$

$$\begin{pmatrix} s_0 & s_1 & \cdots & s_t \\ s_1 & s_2 & \cdots & s_{t+1} \\ \vdots & \vdots & \cdots & \vdots \\ s_{t-1} & s_t & \cdots & s_{2t-1} \end{pmatrix} \begin{pmatrix} \phi_t \\ \phi_{t-1} \\ \vdots \\ \phi_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.6}$$

This approach relies on solving first (3.6) and then substitutes the values of the ϕ_u's in (3.5) to get the values of the ψ_u's.

One can also remark from $\phi_0 = 1$ that (3.6) can be rewritten as:

$$\begin{cases} -s_t & = & \phi_t s_0 & + & \phi_{t-1} s_1 & + & \cdots & + & \phi_1 s_{t-1} \\ -s_{t+1} & = & \phi_t s_1 & + & \phi_{t-1} s_2 & + & \cdots & + & \phi_1 s_t \\ & \vdots & & & & & & & \\ -s_{2t-1} & = & \phi_t s_{t-1} & + & \phi_{t-1} s_t & + & \cdots & + & \phi_1 s_{2t-2} \end{cases} \tag{3.7}$$

Several methods exist for solving any system of the form (3.7). There is the Berlekamp-Massey algorithm which runs in time $\mathcal{O}(t^2)$, and for very large t, the methods developed in [3, 18] which require time $\mathcal{O}(t \log^2 t)$.

The pair (ψ, φ) which is solution of the problem is a Padé approximant of type $(t-1, t)$ of S denoted by $[t-1/t]_S$.This remark, observed by many authors (see [3]), deserves more formalization as follows. For a power series $f(x) = \sum_{i=0}^{\infty} c_i x^i$, define $Ord(f)$ as the least nonnegative integer j such that $c_j \neq 0$, and write in this case $f = \mathbf{O}(x^j)$.

Proposition 3.8. *The system (3.3) is equivalent to the normalized Padé approximant problem*

$$\begin{cases} deg(\varphi) \leq t \\ deg(\psi) \leq t-1 \\ \varphi S - \psi = \boldsymbol{O}(x^{2t}), \end{cases} \tag{3.8}$$

with the normalization $gcd(\varphi, \psi) = 1$, $\varphi(0) = 1$ *and* $s_0 = \psi(0)$. *Furthermore, the normalized solution exists and is unique.*

Proof. Observe by definition, that $\varphi(x)S(x) \equiv \psi(x) \mod x^{2t}$ is equivalent to the existence of a polynomial Q such that $\varphi S - \psi = Q x^{2t}$. We have $Q \neq 0$; otherwise since $gcd(\varphi, \psi) = 1, deg(\psi) < deg(\varphi)$ and φ divides ψ, we get a contradiction. But then $Ord(\varphi S - \psi) \geq 2t$ which is $\varphi S - \psi = \mathbf{O}(x^{2t})$. The existence and uniqueness is a result attributed to Frobenius (see Brent et al. [3].)

□

4. Lattices over Polynomial Rings

There is an analogue notion of lattice in which the ring $\mathbb{Z}$ of integers is replaced by the ring $\mathbb{K}[x]$ where $\mathbb{K}$ is a field. Let us recall that $\deg : \mathbb{K}[x] \to \mathbb{R} \cup \{-\infty\}$ is the map which associates to each nonzero polynomial its degree and to 0 the value $-\infty$.

Definition 4.1 ([13])**.** A $\mathbb{K}[x]$-lattice is a pair consisting of a finitely generated $\mathbb{K}[x]$-module L and a function $q : L \to \mathbb{R} \cup \{-\infty\}$ with the following properties:

(1) $q(X) \neq -\infty$, for all $X \in L, X \neq 0$,
(2) $q(X+Y) \leq max\{q(X), q(Y)\}$ for all $X, Y \in L$,
(3) $q(cX) = \deg(c) + q(X)$ for all $c \in \mathbb{K}[x]$, $X \in L$,
(4) $\dim_{\mathbb{K}}\{X \in L : q(X) \leq r\} < \infty$ for each $r \in \mathbb{R}$.

Note that the second and third properties imply that $\{X \in L : q(X) \leq r\}$ is a $\mathbb{K}$-vector space for each $r \in \mathbb{R}$.

Example 4.2 ([13])**.** For each $\lambda \in \mathbb{R}$, an example of a $\mathbb{K}[x]$-lattice is given by $L = \mathbb{K}[x]$, $q(f) = \lambda + deg(f)$; this lattice is denoted by $\mathcal{O}(-\lambda)$.

If L_1, q_1 and L_2, q_2 are $\mathbb{K}[x]$-lattices, then their orthogonal sum is the $\mathbb{K}[x]$-lattice $L = L_1 \oplus L_2$ with $q(X_1, X_2) = max(q_1(X_1), q_2(X_2))$, for $X_1 \in L_1$, $X_2 \in L_2$.

There exists for every $\mathbb{K}[x]$-lattice a finite sequence $\lambda_1, ..., \lambda_n$ of real numbers such that the lattice is, in an obvious sense, isomorphic to the orthogonal sum of the n lattices $\mathcal{O}(-\lambda_i)$, $1 \leq i \leq n$:

$$L = \oplus_{\lambda_i} \mathcal{O}(-\lambda_i)$$

If we also require $\lambda_1 \leq ... \leq \lambda_n$, then the λ_i are uniquely determined as the successive minima of the lattice, and all λ_i are in $\mathbb{Z}$ if and only if the lattice is integral-valued.

4.1. **Basis reduction.** Let L, q be a $\mathbb{K}[x]$-lattice. Then L has a basis as a $\mathbb{K}[x]$-module. A basis $b_1, \ldots, b_n$ is called *reduced* if for each $(c_i)_{i=1}^n \in \mathbb{K}[x]^n$ one has

$$q\left(\sum_{i=1}^{n} c_i b_i\right) = max\{q(c_i b_i) : 1 \leq i \leq n\}.$$

In the last example, it is easy to see that each $\mathbb{K}[x]$-lattice has a reduced basis (cf. [13]).

A.K Lenstra [12] devised an algorithm to find a reduced basis. He defines the orthogonality defect of a basis $B = (b_1, ..., b_n)$ for a lattice (L, q) as

$$OD(B) = \sum_{i=1}^{n} q(b_i) - q(Det(L)),$$

where $Det(L) = Det(B)$ is the determinant of the lattice (defined up to a unit in $\mathbb{K}$.)

Theorem 4.3 ([12])**.** *There is an algorithm which transforms a basis B into a reduced basis $b'_1, ..., b'_n$ such that $q(b'_1) \leq \cdots \leq q(b'_n)$, in computational time $\mathcal{O}(n^3(OD(B) + 1)max_{1 \leq i \leq n}(q(b_i) + 1))$. Furthermore b'_1 is a shortest element of L.*

4.2. **A characterization theorem by lattices.** Consider the key equation:

$$\varphi(x)S(x) \equiv \psi(x) \bmod x^{2t}. \tag{4.1}$$

and let

$$M = \begin{pmatrix} 1 & S \\ 0 & x^{2t} \end{pmatrix} \tag{4.2}$$

We define the key $\mathbb{F}_q[x]$-lattice by:

$$L_{\mathcal{C}} = \left\{(u_1, u_2)M : (u_1, u_2) \in \mathbb{F}_q[x]^2\right\},$$

$$\forall (\phi_1, \psi_1) \in L_{\mathcal{C}}, q(\phi_1, \psi_1) = max(deg(\phi_1), deg(\psi_1)).$$

Based on the above, we deduce the following important proposition:

Proposition 4.4.

- *The solution set of the key equation is the lattice $L_{\mathcal{C}}$;*

- $L_{\mathcal{C}} = \mathbb{F}_q[x]v_1 \oplus \mathbb{F}_q[x]v_2$ *with* $v_1 = (1, S)$ *and* $v_2 = (0, x^{2t})$

Remark 4.5. The key lattice for a Goppa code $\Gamma(L, g)$ can be defined using the same method, by:

$$L_{\mathcal{C}} = \left\{(u_1, u_2)M : (u_1, u_2) \in \mathbb{F}_q[x]^2\right\}$$

with

$$M = \begin{pmatrix} 1 & S \\ 0 & g(x) \end{pmatrix}$$

Lemma 4.6. *If z is a shortest nonzero vector in $L_{\mathcal{C}}$, then $q(z) \leq t$.*

Proof. This results from the fact that the solution, noted (φ, ψ), of the congruence (3.3) is an element of $L_{\mathcal{C}} - \{0\}$, and $q(\varphi, \psi) \leq t$.

□

The following theorem gives a characterization of the congruence (3.3).

Theorem 4.7. *The solution of the congruence (3.3) is a minimal nonzero vector $z = (\phi_1, \psi_1)$ of the lattice $L_{\mathcal{C}}$ (up to scaling by a unit in $\mathbb{F}_q$) satisfying $deg(\psi_1) < deg(\phi_1)$.*

Proof. Let (ϕ_1, ψ_1) a minimal nonzero vector in $L_{\mathcal{C}}$.

Suppose that $deg(\psi_1) < deg(\phi_1)$. We can deduce from the proposition 4.4 that (ϕ_1, ψ_1) satisfies the key equation, as $deg(\psi_1) < deg(\phi_1)$,and the lemma 4.6 implies that $deg(\phi_1) \leq t$ and $deg(\psi_1) < t$.

Let (φ, ψ) be the solution of the key equation (3.3). The lemma 3.2 implies that there exists a polynomial $\lambda(x) \in \mathbb{F}_q[x]$ such that $\phi_1(x) = \lambda(x)\varphi(x)$ and $\psi_1(x) = \lambda(x)\psi(x)$. Since (ϕ_1, ψ_1) is minimal, $(\varphi, \psi) \in L_{\mathcal{C}}$, and $q(\phi_1, \psi_1) = deg(\lambda)+q(\varphi, \psi)$, then λ is a constant polynomial, which completes the proof.

□

Refining further this theorem permits to eliminate the condition on the *gcd* :

Corollary 4.8. (1) *If $z = (\phi_1, \psi_1)$ is a shortest nonzero vector in $L_{\mathcal{C}}$ with $deg(\psi_1) < deg(\phi_1)$, then $gcd(\phi_1, \psi_1) = 1$.*

(2) *Let (φ, ψ) be the solution of the congruence (3.3) and set $q(\varphi, \psi) = \rho$, then :*

i): *ρ is the smallest integer in $\{0, ..., t\}$ such that $q(z) = \rho$ has a solution in $L_{\mathcal{C}} - \{0\}$.*

ii): *the solution of the system*

$$\begin{cases} q(z) = \rho \\ z \in L_{\mathcal{C}} - \{0\} \\ z = (\phi_1, \psi_1) \\ deg(\psi_1) < deg(\phi_1) \end{cases} \tag{4.3}$$

is (φ, ψ) up to scaling by a unit.

Proof. Let (φ, ψ) be the solution of the congruence (3.3).

To show 1), let $z = (\phi_1, \psi_1)$ be a shortest nonzero vector in $L_{\mathcal{C}}$ with $deg(\psi_1) < deg(\phi_1)$. Theorem 4.7 implies that there exists an unit λ such that $\phi_1 = \lambda\varphi$ and $\psi_1 = \lambda\psi$, hence the result.

Set $q(\varphi, \psi) = \rho$.

To show 2)i), suppose that there exists an integer ρ' in $\{0, ..., t\}$ such that $q(z) = \rho'$ has a solution in $L_{\mathcal{C}} - \{0\}$, denoted (ϕ_1, ψ_1), and $\rho' \leq \rho$, by the theorem 4.7 (φ, ψ) is a minimal nonzero vector of the lattice $L_{\mathcal{C}}$, so $\rho \leq \rho'$ and hence we have $\rho = \rho'$.

The statement 2)ii) results from the theorem 4.7 and the fact that any solution of the system (4.3) is a minimal nonzero vector of the lattice $L_{\mathcal{C}}$.

□

Consider the following optimization problem over lattices:

$$min\{q(z) : z \in L_{\mathcal{C}} - \{0\}\} \ (M_{\mathcal{C}})$$

Theorem 4.9 ([8]). *The problem $(M_{\mathcal{C}})$ is solvable in polynomial time.*

Note that, to the problem $(M_{\mathcal{C}})$, we can also apply Theorem 4.3 and we get an $\mathcal{O}(t^2)$ algorithm since $OD(M) = 2t - 1$ where M is given by expression (4.2).

5. Finding a minimal vector and solving the key equation

5.1. **A first algorithm.** Consider the $t-$decoding problem

$$D_{\mathcal{C}} = \begin{cases} minq(z) \\ z \in L_{\mathcal{C}} - \{0\} \\ z = (\varphi, \psi) \\ deg(\psi) < deg(\varphi) \leq t \end{cases} \tag{5.1}$$

We present an algorithm, which is inspired by the work of Grigor'ev [8, p.1766], for solving the problem $D_{\mathcal{C}}$. The algorithm is based on the characterization given by Theorem 4.7.

We find a vector $u = (u_1, u_2)$ over $\mathbb{F}_q[x]$ such that $uM = (\phi_1, \psi_1)$ is a minimal nonzero vector in the lattice $L_{\mathcal{C}}$ with $deg(\psi_1) < deg(\phi_1)$.

Since $deg(S) < r = 2t$ and $q(uM) \leq t$ by lemma 4.6, then $deg(u_1) \leq t$ and $deg(u_2) < 2t$.

Hence the question of whether it is true that $q(uM) = r'$, for any given $r' \leq t$ reduces to solving a linear system LS over $\mathbb{F}_q$ below, in which the unknowns are the coefficients of the polynomials u_1 , u_2.

The algorithm tests, successively, $r' = 0, 1, ...$ up to the value of r' for which the mentioned system is solvable (such that $r' \leq t$). The solution (ϕ_1, ψ_1) that we seek is one that satisfies $deg(\psi_1) < deg(\phi_1)$.

Let $u_1 = \sum_{k=0}^{t} u_{1,k}x^k$ and $u_2 = \sum_{k=0}^{2t-1} u_{2,k}x^k$. We will denote by $\mathcal{S}_{r'}$ the set of uM such that $u = (u_1, u_2)$ is a solution of the system

$$\begin{cases} q(uM) = r' \\ deg(u_1 S + x^{2t}u_2) < deg(u_1) \end{cases} \tag{5.2}$$

Then any vector $u \in \mathcal{S}_{r'}$ must verify

$$u_{1,r'+1} = \cdots = u_{1,t} = 0 \tag{5.3}$$

$$u_{2,t} = \cdots = u_{2,2t-1} = 0 \tag{5.4}$$

$$\forall k = 0, \ldots, t-1,\ u_{2,k} + \sum_{i+j=k+2t} u_{1,i} s_j = 0 \tag{5.5}$$

$$\forall k = r', \ldots, 2t-1, \sum_{i+j=k} u_{1,i} s_j = 0 \tag{5.6}$$

The justification of (5.3) to (5.6) is as follows. The first component of uM is u_1, hence (5.3). Its second component $u_1 S + u_2 x^{2t}$ may be written as

$$\sum_{k=0}^{3t-1} \Big(\sum_{k=i+j} u_{1,i} s_j \Big) x^k + \sum_{i=0}^{2t-1} u_{2,i} x^{i+2t}$$

$$= \sum_{k=0}^{3t-1} \Big(\sum_{k=i+j} u_{1,i} s_j \Big) x^k + u_{2,0} x^{2t} + \cdots + u_{2,t-1} x^{3t-1}$$

$$+ u_{2,t} x^{3t} + \cdots + u_{2,2t-1} x^{4t-1},$$

which implies (5.4) to (5.6).

Let $\mathcal{S}'_{r'}$ be the set of uM where $u = (u_1, u_2)$ is a solution of the system formed by the equations (5.3) to (5.6). We note that $\mathcal{S}_{r'} \subseteq \mathcal{S}'_{r'}$ and

$$\mathcal{S}'_{r'} \subseteq \mathcal{S}'_{r'+1} \subseteq \cdots \subseteq \mathcal{S}'_{t} \tag{5.7}$$

We restate the following version of theorem 4.7 which gives the correctness of the **algorithm** 1 below, and localize the solution up to a unit.

Theorem 5.1. *Let (φ, ψ) be the solution of the congruence (3.3), and set $q(\varphi, \psi) = \rho$. Then ρ is the smallest integer in $\{0, ..., t\}$ such that*

(1) *the system (5.2) has a solution in $L_{\mathcal{C}}$;*
(2) *the system formed by the equations (5.3) to (5.6) has a nontrivial solution;*

and we have:

$\mathcal{S}_\rho = \mathcal{S}'_\rho - \{0\} = \{\lambda(\varphi, \psi) \ / \ \lambda \in \mathbb{F}_q - \{0\}\}$.

Proof. Corollary 4.8 implies 1) and $\mathcal{S}_\rho = \{\lambda(\varphi, \psi) \ / \ \lambda \in \mathbb{F}_q - \{0\}\}$.

According to the above we have $\mathcal{S}_\rho \subset \mathcal{S}'_\rho - \{0\}$, and from which $\mathcal{S}'_\rho - \{0\} \neq \emptyset$. So the system formed by the equations (5.3) to (5.6) has a nontrivial solution.

Suppose that there exists an integer ρ' in $\{0, ..., t\}$ such that $\rho' \leq \rho$ and the system formed by the equations (5.3) to (5.6) has a nontrivial solution $u = (u_1, u_2)$, since $uM = (\phi_1, \psi_1)$ is an element of $L_{\mathcal{C}} - \{0\}$ then $q(\phi_1, \psi_1) \leq q(\varphi, \psi)$, whence $\rho' = \rho$, which proves 2).

To complete the proof of the theorem, it suffices to show the inclusion $\mathcal{S}'_\rho - \{0\} \subset \mathcal{S}_\rho$.

Let $(u_1, u_2)M = (\phi_1, \psi_1) \in \mathcal{S}'_\rho - \{0\}$, then (ϕ_1, ψ_1) satisfies the key equation, furthermore $deg(\phi_1) = deg(u_1) \leq \rho \leq t$ and $deg(\psi_1) = deg(u_1 S + x^{2t}) \leq \rho - 1 < t$. The lemma 3.2 implies that exists a polynomial λ in $\mathbb{F}_q[x] - \{0\}$ such that $\phi_1(x) = \lambda(x)\varphi(x)$ and $\psi_1(x) = \lambda(x)\psi(x)$.

Since $\rho \leq deg(\lambda) + q(\varphi, \psi) = q(\phi_1, \psi_1) \leq \rho$, we have $q(\phi_1, \psi_1) = q(\varphi, \psi) = \rho$ and $\lambda \in \mathbb{F}_q - \{0\}$, and consequently $(\phi_1, \psi_1) \in \mathcal{S}_\rho$.

□

We summarize the previous discussion in **Algorithm 1** :

Data: M: matrix, t: integer
Result: Solution (φ, ψ) of the congruence (3.3)
We set: $u = (u_1, u_2) \in \mathbb{F}_q[x]^2$;
$u_1 = \sum_{k=0}^{t} u_{1,k} x^k$ and $u_2 = \sum_{k=0}^{2t-1} u_{2,k} x^k$;
(LS) : the system formed by the equations (5.3) to (5.6); $r' \leftarrow 0$;
(1) **while** $r' \leq t$ **do**
  $\mathcal{S} \leftarrow solve(LS)$;
  if $\mathcal{S} \neq \{0\}$ **then**
    Choose $u \in \mathcal{S} - \{0\}$; compute $(\varphi, \psi) = uM$
  else
    $r' \leftarrow r' + 1$
  end
end 2
Apply Forney's formula to (φ, ψ)

Algorithm 1: Decoding with lattices

Remark 5.2. The time complexity of this algorithm is $\mathcal{O}(t \cdot time(LS))$, where $time(LS)$ is the time complexity of solving the problem (LS). We observe that the length of the **while** loop of algorithm 1 may be diminished using a dichotomy principle, taking into account the relation (5.7); this results in an $\mathcal{O}(\log(t) \cdot time(LS))$ algorithm. A second observation is that the structure of the system (LS) is Toeplitz-like (see [3, 18]) and this may improve the complexity of our problem.

The structure of **algorithm 1** suggest to examine the parallel complexity of the problem $D_{\mathcal{C}}$. Recall that NC is the class of problems with fast parallel time $\mathcal{O}(\log^{\mathcal{O}(1)} n)$ and polynomial size $\mathcal{O}(n^{\mathcal{O}(1)})$ where n is the size of the problem [1, 4]. In the following result, the models of computation are algebraic or arithmetic circuits.

Proposition 5.3. *The problem of decoding alternant codes is in the algebraic class NC.*

Proof. (sketch). Observe that the problem (LS) is in algebraic NC. See [1, 4] for more concepts and to fill in details.
The arithmetic circuit A corresponding to Algorithm1, proceeds in three phases (see figure 1). The first phase, 'Duplication Tree', duplicates the input (M, t), whose outputs serve as inputs to the modules $M_1, ..., M_t$. The depth of duplication tree is $\mathcal{O}(log(size(input))$. The second phase consists of modules $M_{r'}$ where each $M_{r'}$ solves $LS_{r'}$ and tests the elements of $\mathcal{S}_{r'}$. The third phase is a Test/Select

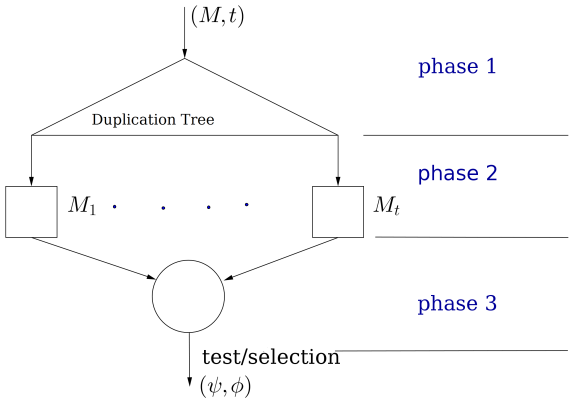


FIGURE 1. Sketch of the parallel version of Algorithm1

receiving its inputs from the $M_{r'}$ and outputs the desired (φ, ψ). The overall size of A is polynomial in t and its depth is polylogarithmic.

□

Example 5.4. Let us illustrate by an example of the alternant code taken from [15, p. 335].

Let β be a primitive element of $\mathbb{F}_8$, $\beta^3 = \beta + 1$.

Set $\alpha = (1, \beta, \beta^2, ..., \beta^6)$ and $y = (1, 1, 1, ..., 1)$.

The alternant code $\mathcal{C} = \mathcal{A}(\alpha, y, 2)$ has a parity check matrix equal to

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

Remplacing each entry by the corresponding binary vector of length 3, the code has a parity check matrix of rank 4 over $\mathbb{F}_2$:

$$\tilde{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$\mathcal{C}$ is a $[7, 3, 4]$-code, $d = 4 > 3 = r + 1$ and $t = 1$. The tuple $c = (0, 1, 1, 1, 0, 0, 1)$ is a codeword of $\mathcal{C}$, and $v = c + e = (1, 1, 1, 1, 0, 0, 1)$ is the received word where $e = (1, 0, 0, 0, 0, 0, 0)$ is the error vector.

The syndrome vector $S = vH^\top = (1, 1, 0, 0)$, and the polynomial syndrome is $S(x) = 1 + x$.

We have $M = \begin{pmatrix} 1 & 1+x \\ 0 & x^2 \end{pmatrix}$, the lattice which corresponds to M is :

$$\begin{aligned} L_{\mathcal{C}} &= \{(u_1, u_2)M : (u_1, u_2) \in \mathbb{F}_2[x]^2\} \\ &= \{(u_1, (1+x)u_1 + x^2 u_2) : (u_1, u_2) \in \mathbb{F}_2[x]^2\} \end{aligned}$$

We search a vector $u = (u_1, u_2)$ over $\mathbb{F}_2[x]$ such that uM is a minimal vector in the lattice $L_{\mathcal{C}}$.

First, we note that $\deg(u_1) \leq t$ and $\deg(u_2) < 2t$.

Set $u_1(x) = u_{1,0} + u_{1,1}x$ and $u_2(x) = u_{2,0} + u_{2,1}x$.

The algorithm tests, successively, $r' = 0, 1$, up to the value of r' for which the linear system (LS) mentioned above has a nontrivial solution (knowing that $r' \leq 1$).

For $r' = 0$:

The associated linear system

$$\begin{cases} u_{1,1} = 0 \\ u_{2,1} = 0 \\ u_{2,0} + u_{1,1}s_1 = 0 \\ u_{1,0}s_0 = 0 \\ u_{1,1}s_1 = 0 \end{cases}$$

admits a trivial solution $u = (0, 0)$.

For $r' = 1$: the associated linear system (LS) is as follows

$$\begin{cases} u_{2,1} = 0 \\ u_{2,0} + u_{1,1}s_1 = 0 \\ u_{1,0}s_0 + u_{1,1}s_0 = 0 \end{cases}$$

which is equivalent to

$$\begin{cases} u_{2,1} = 0 \\ u_{2,0} = u_{1,1} = u_{1,0} \end{cases}$$

Therefore the set of solutions of the system (LS) is $\mathcal{S} = \{(0, 0), (1 + x, 1)\}$.

For $u = (1 + x, 1) \neq (0, 0)$ we have $(\varphi, \psi) = uM = (1 + x, 1)$.

The error location $= 1$, as we are in the binary case, without applying the Forney's formula, just replace the first coordinate of v by its complement, the vector error is $e = (1, 0, 0, 0, 0, 0, 0)$, and the recovered codeword is $c = v + e = (0, 1, 1, 1, 0, 0, 1)$.

5.2. **A $t(\log t)^3$ algorithm.** An inspection of the equations (5.3) to (5.6) reveals that this problem reduces to solving the Toeplitz-like system L_ρ for $\rho = 0, ..., t$:

$$T(s) \cdot \begin{pmatrix} u_{1,0} \\ \vdots \\ u_{1,\rho-1} \end{pmatrix} = 0$$

where

$$T(s) = \begin{pmatrix} s_{\rho-1} & \cdots & s_0 \\ \vdots & \cdots & \vdots \\ s_{2t-1} & \cdots & s_{2t-\rho} \end{pmatrix}$$

and $s = (s_0, ..., s_{2t-1})$ is the syndrome sequence. Note that u_2 is given in terms of u_1 by equations (5.5).

Remark 5.5. Note that the form of L_ρ already bears resemblance to equation (3.5) of section 3.

Definition 5.6. For any sequence $a = (a_k)$ of complex numbers, we define the $m \times n$ Toeplitz matrix by

$$T^{(l)}_{m,n}(a) = \begin{pmatrix} a_l & \dots & a_{l-n+1} \\ \vdots & \dots & \vdots \\ a_{l+m-1} & \dots & a_{l+m-n} \end{pmatrix}$$

When $m = n$, we denote the square matrix by $T^{(l)}_m$.

Lemma 5.7. *There is an algorithm for solving the system L_ρ in time $\mathcal{O}(t(\log t)^2)$.*

Proof. Let $L = \lfloor \frac{2t-\rho+1}{\rho} \rfloor$. Then we have

$$T(s) = \begin{pmatrix} T^{(\rho-1)}_\rho(s) \\ \vdots \\ T^{(L\rho-1)}_\rho(s) \\ T'(s) \end{pmatrix}$$

where $T'(s)$ has a number of rows less than or equal to ρ . We know that solving $T^{(l)}_\rho$ for any l may be done in $\mathcal{O}((\rho)(\log(\rho))^2)$ [3, 18]. Then solving the L systems, testing for the subsystem given by $T'(s)$, and since we have $\rho \leq t$, an upper bound of solving L_ρ is

$$\mathcal{O}(L(\rho)(\log(\rho))^2) = \mathcal{O}(t(\log t)^2).$$

□

Proposition 5.8. *There is an algorithm for solving the problem $D_{\mathcal{C}}$ in time $\mathcal{O}(t(\log t)^3)$.*

Proof. The dichotomy principle is applied to **Algorithm 1** by taking into account the relation (5.7) (or $KerL_\rho \subseteq KerL_{\rho+1}$). We introduce the following recursive procedure $R(a, b)$ for $0 \leq a \leq b \leq t$. The procedure $Solve(L_\rho)$ returns the set of solutions of the system L_ρ as given by lemma 5.7.

- $R(a, a) = Solve(L_a)$.
- $R(a, a + 1) = Solve(L_a)$ if $Solve(L_a) \neq 0$, else $Solve(L_{a+1})$.
- For $b > a + 1$,
  $R(a, b) = R(a, \lfloor \frac{a+b}{2} \rfloor)$ if $Solve(L_{\lfloor \frac{a+b}{2} \rfloor} \neq 0)$, else $R(\lfloor \frac{a+b}{2} \rfloor + 1, b)$

Since the depth of recursion of R is at most $\log t$, we have the result (see **Algorithm 2**). □

Data: M: matrix, t: integer
Result: Solution (φ, ψ) of the congruence (3.3)
We set: $u = (u_1, u_2) \in \mathbb{F}_q[x]^2$;
$u_1 = \sum_{k=0}^{t} u_{1,k}x^k$ and $u_2 = \sum_{k=0}^{2t-1} u_{2,k}x^k$;
(1)$S = R(0, t)$;
(2) Choose $u \in \mathcal{S} - \{0\}$; compute $(\varphi, \psi) = uM$
(3) Apply Forney's formula to (φ, ψ)

Algorithm 2: Decoding with lattices : second version

6. Conclusion

We introduced a method based on the key equation and defined the concept of key lattice to give a conceptually simple decoding. The complexity of the presented algorithm is tractable and reduces to solving Toeplitz-like systems. It is also possible to investigate the case of alternant codes over finite commutative rings (see [17]).

The case of Goppa codes on algebraic curves, which generalize rational Goppa code is an interesting line of research (see [14, 21] and the references cited in these papers).

7. ACKNOWLEDGMENT

The authors would like to thank Sidi Mohamed Ben Abdellah Univercity (USMBA), LSI and FP of Taza in MOROCCO for their valued supports.

References

[1] Borodin, A., von zur Gathen,J., Hopcroft, J. E.: Fast Parallel Matrix and GCD Computations. Information and Control 52(3): 241-256 (1982).
[2] Bostin, A., Algorithmes rapides pour les polynômes, séries formelles et matrices. Journées Nationales de Calcul Formel. CIRM (Luminy Marseille) 2010.
[3] Brent,R. P., Gustavson, F. G., Yun, D. Y. Y., Fast solution of Toeplitz systems of equations and computation of Padé approximants. J. Algorithms 1 (1980), 259-295.
[4] Burgisser, P., Clausen, M., Shokrollahi, M. A., Algebraic complexity theory. Springer,1997.
[5] Demazure, M., Cours d'algèbre. Nouvelle bibliothèque mathématique. Cassini, 2008.
[6] Fitzpatrick, P., On the Key Equation, IEEE Trans. Inform. Theory 41 (1995), pp. 1290-1302.
[7] Goppa, V. D., Algebraico-geometric codes. Math. USSR Izvestiya, Vol. 21 (1983), No. 1, pp. 75-91.
[8] Grigor'ev, D. Yu., Factorization of polynomials over a finite field and the solution of systems of algebraic equations. Plenum Publishing Corporation, 1986.
[9] Huffman, W.C , Pless, V., Fundamentals of Error Correcting Codes, Cambridge University Press, 2003.
[10] Joachim von zur Gathen. Hensel and Newton methods in valuation rings. Math. Comp. 42 (1984), no. 166, pp. 637-661.
[11] J. von zur Gathen and Jurgen Gerhard. Modern Computer Algebra. Second Edition 2003. Cambridge.
[12] Lenstra, A. K., Factoring Multivariate Polynomials over finite fields. JCSC, Volume 30, pp 235-248, 1985.

[13] Lenstra, H. W., Lattices. Algorithmic Numer Theory. MSRI Publications. Volume 44, 2008.
[14] Little, J. B., A key equation and the computation of error values for codes from order domains. See http://arxiv.org/abs/math/0303299v2
[15] Mac Williams,F. J., Sloane, N.J.A., The Theory of Error-Correcting Codes, north-holland publishing company Amsterdam New York Oxford. Third printing 1981.
[16] McEliece, R. J., and James B. Shearer, A Property of Euclid's Algorithm and an Application to Pade Approximation. SIAM Journal on Applied Mathematics, Vol. 34, No. 4 (Jun., 1978), pp. 611-615.
[17] Norton, G. H., Salagean,A., On Efficient Decoding of Alternant Codes over a Commutative Ring. IMA Conference on Cryptography and Coding , pp. 173-178, 1999.
[18] Pan, P., Structured Matrices and Polynomials, Unified Superfast Algorithms. Birkhäuser, Boston, 2001.
[19] Patterson, N., The algebraic decoding of Goppa codes. Information Theory, IEEE Transactions on, 21(2):203 - 207, March 1975.
[20] Sebastià Xambó i Descamps, A computational primer on block error-correcting codes. January 2003. Facultat de Matematiques I Estadística. http://www-ma2.upc.edu/sxd/CC/CC-Book.html.
[21] Sullivan, M. O., The key equation for one-point codes and efficient error evaluation, J. Pure Appl. Algebra 169 (2002), 295-320.
[22] van Lint, J. H., Introduction to Coding Theory. Third Edition , Springer.

[1]CRMEF-FES, Morocco.
E-mail address: abkhalid9@gmail.com

[2]Moulay Ismaïl University, ENSAM-Meknès, Morocco.
E-mail address: hbenazza@yahoo.com
Partially supported by " PHC Informatique théorique au maghreb" project.

[3] University of Rouen, France.
E-mail address: ayoub.otmani@univ-rouen.fr
Partially supported by " PHC Informatique théorique au maghreb" project.