The decoding of algebraic geometry codes

Peter Beelen and Tom Høholdt

Department of Mathematics,
Technical University of Denmark,
Matematiktorvet, Building 303S,
DK 2800, Kgs.Lyngby,
{p.beelen,t.hoeholdt}@mat.dtu.dk

4. juli 2008

Contents

Introduction

- The work on decoding of algebraic geometry codes started in 1986 and in the following 10 years a lot of papers appeared. In the Handbook on Coding Theory The paper all (or most of) the work on decoding until 1997 is surveyed.
- These lectures present decoding algorithms using recent ideas and methods.
  - The *basic* algorithm for decoding general algebraic geometry codes
  - Syndrome formulation of the basic algorithm
  - Generalized order bound and majority voting
  - List decoding
  - Syndrome formulation of list decoding

Contents

Decoding

- When an (n, k) code C is used for correcting errors, one of the important problems is the design of a *decoder*.

Decoding

- When an (n, k) code C is used for correcting errors, one of the important problems is the design of a *decoder*.
- A decoder is ?

Decoding

- When an (n, k) code C is used for correcting errors, one of the important problems is the design of a *decoder*.
- A decoder is ?
- One way of stating the objective of the decoder is: for a received vector r, select a codeword c that minimizes $d(r, c)$. This is called *maximum likelihood decoding*. It is clear that if the code is t-error correcting, i.e $t < \frac{d_{min}}{2}$ and $r = c + e$ with $w(e) \leq t$ then the output of such a decoder is c.

Decoding

- When an (n, k) code C is used for correcting errors, one of the important problems is the design of a *decoder*.
- A decoder is ?
- One way of stating the objective of the decoder is: for a received vector r, select a codeword c that minimizes $d(r, c)$. This is called *maximum likelihood decoding*. It is clear that if the code is t-error correcting, i.e $t < \frac{d_{min}}{2}$ and $r = c + e$ with $w(e) \leq t$ then the output of such a decoder is c.
- It is often difficult to design a maximum likelihood decoder, but if we only want to correct t errors where $t < \frac{d_{min}}{2}$ it is sometimes easier to get a good algorithm.

Minimum distance and list decoders

Definition

A *minimum distance decoder* is a decoder that, given a received word r, selects the codeword c that satisfies $d(r,c) < \frac{d_{min}}{2}$ if such a codeword exists, and otherwise declares failure.

Minimum distance and list decoders

Definition

A *minimum distance decoder* is a decoder that, given a received word r, selects the codeword c that satisfies $d(r,c) < \frac{d_{min}}{2}$ if such a codeword exists, and otherwise declares failure.

We will also in the following consider a so-called *list decoder*

Definition

Let $0 \leq \tau \leq n$. A τ list decoder is a decoder that, given a received word r, outputs all codewords c such that $d(r,c) \leq \tau$.

Minimum distance and list decoders

Definition

A *minimum distance decoder* is a decoder that, given a received word r, selects the codeword c that satisfies $d(r,c) < \frac{d_{min}}{2}$ if such a codeword exists, and otherwise declares failure.

We will also in the following consider a so-called *list decoder*

Definition

Let $0 \leq \tau \leq n$. A τ list decoder is a decoder that, given a received word r, outputs all codewords c such that $d(r,c) \leq \tau$.

If $\tau < \frac{d_{min}}{2}$ then there is at most one codeword, but for larger τ there could be more, hence the name list decoder. For practical purposes, the list should be small.

The basic algorithm

- Let χ be an algebraic curve, i.e. an absolutely irreducible and nonsingular affine or projective variety of dimension one, whose defining equations are (homogeneous) polynomials with coefficients in a finite field $\mathbb{F}$.

The basic algorithm

- Let χ be an algebraic curve, i.e. an absolutely irreducible and nonsingular affine or projective variety of dimension one, whose defining equations are (homogeneous) polynomials with coefficients in a finite field $\mathbb{F}$.
- Let $\mathscr{F}$ and g denote the function field and genus of χ respectively.

The basic algorithm

- Let χ be an algebraic curve, i.e. an absolutely irreducible and nonsingular affine or projective variety of dimension one, whose defining equations are (homogeneous) polynomials with coefficients in a finite field $\mathbb{F}$.
- Let $\mathscr{F}$ and g denote the function field and genus of χ respectively.
- Let G and $D = P_1 + \cdots + P_n$ be $\mathbb{F}$-rational divisors on χ with $\operatorname{supp} D \cap \operatorname{supp} G = \varnothing$.

The basic algorithm

- Let χ be an algebraic curve, i.e. an absolutely irreducible and nonsingular affine or projective variety of dimension one, whose defining equations are (homogeneous) polynomials with coefficients in a finite field $\mathbb{F}$.
- Let $\mathscr{F}$ and g denote the function field and genus of χ respectively.
- Let G and $D = P_1 + \cdots + P_n$ be $\mathbb{F}$-rational divisors on χ with $\operatorname{supp} D \cap \operatorname{supp} G = \varnothing$.
- Define the functions

$$\mathrm{Ev}_D : L(G) \to \mathbb{F}^n, \quad f \mapsto (f(P_1), \ldots, f(P_n))$$

$$\mathrm{Res}_D : \Omega(G-D) \to \mathbb{F}^n, \quad \omega \mapsto (\mathrm{res}_{P_1}(\omega), \ldots, \mathrm{res}_{P_n}(\omega))$$

that are used to construct the codes $C_L(D, G)$ and $C_\Omega(D, G)$.

Interpolation polynomial

- We wish to decode $C_L(D, G)$. Say we have received the word $(r_1, \ldots, r_n)$ containing at most t errors.

Interpolation polynomial

- We wish to decode $C_L(D, G)$. Say we have received the word $(r_1, \ldots, r_n)$ containing at most t errors.
- The idea of the algorithm is to find an *interpolation polynomial* $Q(y) \in \mathscr{F}[y] \setminus \{0\}$, such that:
  - (i) $Q(y) = Q_0 + Q_1 y$ where $Q_0 \in L(A)$ and $Q_1 \in L(A - G)$
  - (ii) $Q_0(P_j) + r_j Q_1(P_j) = 0, j = 1, \ldots, n$

- The basic algorithm works with a divisor A with $\mathrm{supp}\, A \cap \mathrm{supp}\, D = \varnothing$ satisfying
  1. $\deg A < n - t$
  2. $\deg A > \frac{n+\deg G}{2} + g - 1$

- The basic algorithm works with a divisor A with $\operatorname{supp} A \cap \operatorname{supp} D = \varnothing$ satisfying
  1. $\deg A < n - t$
  2. $\deg A > \frac{n+\deg G}{2} + g - 1$
- If $t < \frac{n-\deg G}{2} - g$ one can show that such a divisor A exists. We will see later that condition (2) can be relaxed and then we can work with larger t.

Interpolation polynomial

Lemma

Suppose the transmitted word is $ev_D(f)$ *with* $f \in L(G)$ *and* $Q(y)$ *satisfy* (i) *and* (ii) *then* $f = -\frac{Q_0}{Q_1}$

Interpolation polynomial

Lemma

Suppose the transmitted word is $ev_D(f)$ with $f \in L(G)$ and $Q(y)$ satisfy (i) and (ii) then $f = -\frac{Q_0}{Q_1}$

Since $f \in L(G)$ and $Q_1 \in L(A - G)$ we have $fQ_1 \in L(A)$ and hence $Q(f) \in L(A)$. We also have

$$Q_0(P) + f(P)Q_1(P) = 0,$$

for at least $n - t$ of the points P in $\{P_1, \ldots, P_n\}$, so $Q(f)$ is in $L(A - P_{i_1} - \cdots - P_{i_s})$ where $s \geq n - t$. But $\deg(A - P_{i_1} - \cdots - P_{i_s}) < 0$ and therefore $Q(f) = 0$ and the result follows. □

Existence of $Q(y)$

Remark

Note that $Q(y) = Q_1 \cdot (y - f)$ and thus Q_1 must have the error-positions among its zeroes. Hence Q_1 is called an error-locator.

Existence of $Q(y)$

Remark

Note that $Q(y) = Q_1 \cdot (y - f)$ and thus Q_1 must have the error-positions among its zeroes. Hence Q_1 is called an error-locator.

Lemma

If the divisor A satisfies condition (2) above then there exists a nonzero $Q(y) \in \mathscr{F}[y]$ satisfying (i) and (ii).

Existence of $Q(y)$

Remark

Note that $Q(y) = Q_1 \cdot (y - f)$ and thus Q_1 must have the error-positions among its zeroes. Hence Q_1 is called an error-locator.

Lemma

If the divisor A satisfies condition (2) above then there exists a nonzero $Q(y) \in \mathscr{F}[y]$ satisfying (i) and (ii).

Let $\{g_1, \ldots, g_{l_0}\}$ be a basis for $L(A)$ and $\{h_1, \ldots, h_{l_1}\}$ a basis for $L(A - G)$. We then write

$$Q_0 = \sum_{i=1}^{l_0} q_{0i} g_i \text{ and } Q_1 = \sum_{i=1}^{l_1} q_{1i} h_i$$

Existence of $Q(y)$

so (ii) becomes

$$\sum_{i=1}^{l_0} q_{0i} g_i(P_j) + r_j \sum_{i=1}^{l_1} q_{1i} h_i(P_j) = 0, \text{ with } j = 1, \ldots, n.$$

Existence of $Q(y)$

so (ii) becomes

$$\sum_{i=1}^{l_0} q_{0i} g_i(P_j) + r_j \sum_{i=1}^{l_1} q_{1i} h_i(P_j) = 0, \text{ with } j = 1, \ldots, n.$$

Since $l_0 + l_1 = l(A) + l(A - G) \geq \deg A + \deg(A - G) - 2g + 2 = 2 \deg A - \deg G - 2g + 2 > n$ the n linear homogenous equations have more that n unknowns (q_{0i} and q_{1i}) so there is a nonzero solution. $\square$

The basic algorithm in pseudo code

Based on the considerations above we can now present the so-called *basic algorithm*:

Input: A received word $(r_1, r_2, \ldots, r_n)$.
Find a polynomial $Q(y)$ satisfying (i) and (ii).
If $f = -\frac{Q_0}{Q_1} \in L(G)$ then
 Output: $\mathrm{Ev}_D(f)$.
Else
 Output: Failure.

So the basic algorithm in this formulation only corrects up to

$$\frac{d}{2} - g \text{ errors.}$$

In specific situations one has to determine the divisor A.

Contents

Syndrome formulation of the basic algorithm

- Reformulation of the basic algorithm using *syndromes*.

Syndrome formulation of the basic algorithm

- Reformulation of the basic algorithm using *syndromes*.
- Easier to find an interpolation polynomial, since its defining system of linear equations can be reduced.

Syndrome formulation of the basic algorithm

- Reformulation of the basic algorithm using *syndromes*.
- Easier to find an interpolation polynomial, since its defining system of linear equations can be reduced.
- Also, the basic algorithm for $C_L(D, G)$ can correct up to $t < (n - \deg G - g)/2$ errors, using syndromes.

Syndrome formulation of the basic algorithm

- Reformulation of the basic algorithm using *syndromes*.
- Easier to find an interpolation polynomial, since its defining system of linear equations can be reduced.
- Also, the basic algorithm for $C_L(D, G)$ can correct up to $t < (n - \deg G - g)/2$ errors, using syndromes.

We introduce matrices:

$$\mathbf{M}_A := \begin{pmatrix} g_1(P_1) & \dots & g_{l_0}(P_1) \\ \vdots & & \vdots \\ g_1(P_n) & \dots & g_{l_0}(P_n) \end{pmatrix}, \qquad (1)$$

Towards syndromes - structured matrices

$$\mathbf{D_r} := \begin{pmatrix} r_1 & & \\ & \ddots & \\ & & r_n \end{pmatrix} \tag{2}$$

and

$$\mathbf{M}_{A-G} := \begin{pmatrix} h_1(P_1) & \dots & h_{l_1}(P_1) \\ \vdots & & \vdots \\ h_1(P_n) & \dots & h_{l_1}(P_n) \end{pmatrix} \tag{3}$$

The interpolation conditions can then be written as:

$$\mathbf{M}_A \cdot \mathbf{q_0} + \mathbf{D_r}\mathbf{M}_{A-G} \cdot \mathbf{q_1} = \mathbf{0}. \tag{4}$$

Reducing the linear system

The system (4) can be solved faster by multiplying from the left with a suitable invertible matrix. We will construct this matrix using differentials on the curve χ.

Reducing the linear system

The system (4) can be solved faster by multiplying from the left with a suitable invertible matrix. We will construct this matrix using differentials on the curve χ.

Lemma

Let A be a non-trivial divisor and write $l_0 = l(A)$. Further let $D = P_1 + \cdots + P_n$ and suppose that $\operatorname{supp} A \cap \operatorname{supp} D = \varnothing$. Then there exists differentials $\omega_1, \ldots, \omega_n$ such that

(i) *The set $\{\operatorname{Res}_D(\omega_1), \ldots, \operatorname{Res}_D(\omega_n)\}$ is a basis for $\mathbb{F}^n$,*

(ii) *The set $\{\operatorname{Res}_D(\omega_1), \ldots, \operatorname{Res}_D(\omega_{n-l_0})\}$ is a basis of $C_\Omega(D, A)$,*

(iii) *For all $P \in \operatorname{supp} D$ and $1 \leq i \leq n$, we have $v_P(\omega_i) \geq -1$,*

(iv) *For any $\mathbf{c} \in C_L(D, A)$ and $1 \leq j \leq n - l_0$, we have $\langle \mathbf{c}, \operatorname{Res}_D(\omega_j) \rangle = 0$.*

Reducing the linear system

Proof:
Take some point T outside $\operatorname{supp} D$ (not necessarily rational). Note that $C_\Omega(D, -T) = \mathbb{F}^n$, since it is the dual of the code $C_L(D, -T)$ and $L(-T) = \{0\}$.

Reducing the linear system

Proof:
Take some point T outside $\mathrm{supp}\, D$ (not necessarily rational). Note that $C_\Omega(D, -T) = \mathbb{F}^n$, since it is the dual of the code $C_L(D, -T)$ and $L(-T) = \{0\}$. So for any $v \in \mathbb{F}^n$, there exists a differential $\omega \in \Omega(-T - D)$ such that $(\mathrm{Res}_D(\omega)) = v$. Since $\deg A < n$, we see that $\dim \Omega(A - D) \geq \dim C_\Omega(D, A) = n - \dim C_L(D, A) = n - l(A) = n - l_0$.

Reducing the linear system

Proof:
Take some point T outside $\operatorname{supp} D$ (not necessarily rational). Note that $C_\Omega(D,-T) = \mathbb{F}^n$, since it is the dual of the code $C_L(D,-T)$ and $L(-T) = \{0\}$. So for any $v \in \mathbb{F}^n$, there exists a differential $\omega \in \Omega(-T-D)$ such that $(\mathrm{Res}_D(\omega)) = v$. Since $\deg A < n$, we see that $\dim \Omega(A-D) \geq \dim C_\Omega(D,A) = n - \dim C_L(D,A) = n - l(A) = n - l_0$. Therefore, starting with a basis $v_1, \ldots, v_{n-l_0}$ of $C_\Omega(D,A)$, we can find differentials $\omega_1, \ldots, \omega_{n-l_0} \in \Omega(A-D)$ such that $v_i = \mathrm{res}(\omega_i)$.

Reducing the linear system

Proof:
Take some point T outside $\mathrm{supp}\, D$ (not necessarily rational). Note that $C_\Omega(D, -T) = \mathbb{F}^n$, since it is the dual of the code $C_L(D, -T)$ and $L(-T) = \{0\}$. So for any $v \in \mathbb{F}^n$, there exists a differential $\omega \in \Omega(-T - D)$ such that $(\mathrm{Res}_D(\omega)) = v$. Since $\deg A < n$, we see that $\dim \Omega(A - D) \geq \dim C_\Omega(D, A) = n - \dim C_L(D, A) = n - l(A) = n - l_0$. Therefore, starting with a basis $v_1, \ldots, v_{n-l_0}$ of $C_\Omega(D, A)$, we can find differentials $\omega_1, \ldots, \omega_{n-l_0} \in \Omega(A - D)$ such that $v_i = \mathrm{res}(\omega_i)$. We can complete the set $\{v_1, \ldots, v_{n-l_0}\}$ to a basis of $\mathbb{F}^n$ by adding l_0 suitable vectors to it, say $\{v_{n-l_0+1}, \ldots, v_n\}$. By the above remark we can then find differentials $\omega_{n-l_0+1}, \ldots, \omega_n \in \Omega(-T - D)$ such that $v_j = \mathrm{Res}_D(\omega_j)$ for all j between $n - l_0 + 1$ and n. This proves items (i), (ii) and (iii).

Reducing the linear system

Proof:
Take some point T outside $\mathrm{supp}\, D$ (not necessarily rational). Note that $C_\Omega(D, -T) = \mathbb{F}^n$, since it is the dual of the code $C_L(D, -T)$ and $L(-T) = \{0\}$. So for any $v \in \mathbb{F}^n$, there exists a differential $\omega \in \Omega(-T - D)$ such that $(\mathrm{Res}_D(\omega)) = v$. Since $\deg A < n$, we see that $\dim \Omega(A - D) \geq \dim C_\Omega(D, A) = n - \dim C_L(D, A) = n - l(A) = n - l_0$. Therefore, starting with a basis $v_1, \ldots, v_{n-l_0}$ of $C_\Omega(D, A)$, we can find differentials $\omega_1, \ldots, \omega_{n-l_0} \in \Omega(A - D)$ such that $v_i = \mathrm{res}(\omega_i)$. We can complete the set $\{v_1, \ldots, v_{n-l_0}\}$ to a basis of $\mathbb{F}^n$ by adding l_0 suitable vectors to it, say $\{v_{n-l_0+1}, \ldots, v_n\}$. By the above remark we can then find differentials $\omega_{n-l_0+1}, \ldots, \omega_n \in \Omega(-T - D)$ such that $v_j = \mathrm{Res}_D(\omega_j)$ for all j between $n - l_0 + 1$ and n. This proves items (i), (ii) and (iii). It is clear that if $j \leq n - l_0$ and $\mathbf{c} \in C_L(D, A)$, then $\langle \mathbf{c}, \mathrm{Res}_D(\omega_j) \rangle = 0$, since $\mathrm{Res}_D(\omega_j) \in C_\Omega(D, A) = C_L(D, A)^\perp$. This proves item (iv), and the lemma follows.

Syndromes

Definition

Let G and $D = P_1 + \cdots + P_n$ be divisors defining a code as usual. Given a differential ω, a function h, and a word $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{F}^n$, we define the following *syndrome*:

$$s_{\omega,h}(\mathbf{r}) := \langle \mathbf{r}, \mathrm{Res}_D(h\omega) \rangle.$$

Syndromes

Definition

Let G and $D = P_1 + \cdots + P_n$ be divisors defining a code as usual. Given a differential ω, a function h, and a word $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{F}^n$, we define the following *syndrome*:

$$s_{\omega,h}(\mathbf{r}) := \langle \mathbf{r}, \mathrm{Res}_D\,(h\omega) \rangle.$$

The name syndrome is justified in the following sense. If $\omega \in \Omega(A - D)$, $h \in L(A - G)$, and $\mathbf{c} = \mathrm{Ev}_D\,(f) \in C_L(D, G)$, then

$$s_{\omega,h}(\mathbf{c}) = \langle \mathrm{Ev}_D\,(f), \mathrm{Res}_D\,(h\omega) \rangle = \sum_{i=1}^{n} f(P_i)\,\mathrm{res}_{P_i}(h\omega) = \sum_{i=1}^{n} \mathrm{res}_{P_i}(fh\omega) \overset{\text{(res. thm.)}}{=} 0$$

Properties of syndromes

Proposition

Let G, D and A be as above, let $\{h_1, \dots, h_{l_1}\}$ be a basis of $L(A-G)$, and let $\omega_1, \dots, \omega_{n-l_0} \in \Omega(A-D)$ be such that $\{\mathrm{Res}_D(\omega_1), \dots, \mathrm{Res}_D(\omega_{n-l_0})\}$ is a basis of $C_\Omega(D, A)$. Then the system (4) is equivalent to:

$$\begin{pmatrix} s_{\omega_1,h_1}(\mathbf{r}) & \dots & s_{\omega_1,h_{l_1}}(\mathbf{r}) \\ \vdots & & \vdots \\ s_{\omega_{n-l_0},h_1}(\mathbf{r}) & \dots & s_{\omega_{n-l_0},h_{l_1}}(\mathbf{r}) \end{pmatrix} \begin{pmatrix} q_{11} \\ \vdots \\ q_{1l_1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (5)$$

The tuple $(q_{11}, \dots, q_{1l_1})$ is a solution of (5) iff there exists a (unique) solution of (4) of the form $(q_{01}, \dots, q_{0l_0}; q_{11}, \dots, q_{1l_1})$.

Properties of syndromes

Proof:

- Let $\omega_1, \ldots, \omega_n$ be differentials satisfying the properties in Lemma 5.

Properties of syndromes

Proof:

- Let $\omega_1, \dots, \omega_n$ be differentials satisfying the properties in Lemma 5.
- From this basis, we define the matrix **H** by putting the i-th row of M equal to $\mathrm{Res}_D(\omega_i)$. We will multiply system (4) with **H** from the left.

Properties of syndromes

Proof:

- Let $\omega_1, \ldots, \omega_n$ be differentials satisfying the properties in Lemma 5.
- From this basis, we define the matrix **H** by putting the i-th row of M equal to $\mathrm{Res}_D(\omega_i)$. We will multiply system (4) with **H** from the left.
- **H** is regular, implying that the multiplied system has exactly the same solutions as the original one.

Properties of syndromes

Proof:

- Let $\omega_1, \ldots, \omega_n$ be differentials satisfying the properties in Lemma 5.
- From this basis, we define the matrix $\mathbf{H}$ by putting the i-th row of M equal to $\mathrm{Res}_D(\omega_i)$. We will multiply system (4) with $\mathbf{H}$ from the left.
- $\mathbf{H}$ is regular, implying that the multiplied system has exactly the same solutions as the original one.
- Since $\deg A < n$, we see that $\dim C_L(D, A) = l(A) = l_0$. Hence the matrix $\mathbf{M}_A$ (and $\mathbf{H}\,\mathbf{M}_A$) has rank l_0.

Properties of syndromes

Proof:

- Let $\omega_1, \ldots, \omega_n$ be differentials satisfying the properties in Lemma 5.
- From this basis, we define the matrix **H** by putting the i-th row of M equal to $\mathrm{Res}_D(\omega_i)$. We will multiply system (4) with **H** from the left.
- **H** is regular, implying that the multiplied system has exactly the same solutions as the original one.
- Since $\deg A < n$, we see that $\dim C_L(D, A) = l(A) = l_0$. Hence the matrix $\mathbf{M}_A$ (and $\mathbf{H}\,\mathbf{M}_A$) has rank l_0.
- On the other hand, according to item 4 in Lemma 5, the first $n - l_0$ rows of $\mathbf{H}\,\mathbf{M}_A$ are zero. Thus the $l_0 \times l_0$ matrix **B** obtained by deleting the first $n - l_0$ rows from $\mathbf{H}\,\mathbf{M}_A$ is regular.

Properties of syndromes

Proof continued:
We have now shown that when we multiply system (4) from the left by $\mathbf{H}$, we obtain a system of the form:

$$\left(\frac{\mathbf{0}}{\mathbf{B}}\right)\begin{pmatrix} q_{01} \\ \vdots \\ q_{0l_0} \end{pmatrix} + \mathbf{H}\,\mathbf{D_r}\,\mathbf{M}_{A-G}\begin{pmatrix} q_{11} \\ \vdots \\ q_{1l_1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \qquad (6)$$

Properties of syndromes

Proof continued:
We have now shown that when we multiply system (4) from the left by $\mathbf{H}$, we obtain a system of the form:

$$\left(\frac{\mathbf{0}}{\mathbf{B}}\right)\begin{pmatrix} q_{01} \\ \vdots \\ q_{0l_0} \end{pmatrix} + \mathbf{H}\,\mathbf{D_r}\,\mathbf{M}_{A-G}\begin{pmatrix} q_{11} \\ \vdots \\ q_{1l_1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \qquad (6)$$

A direct computation shows that the entries of the matrix $\mathbf{H}\,\mathbf{D_r}\,\mathbf{M}_{A-G}$ indeed are syndromes as defined in Definition 6. In other words: system (5) is nothing but the first $n - l_0$ equations of system (6). Since $\mathbf{B}$ is regular, the claim of the proposition now follows. $\square$

Syndrome matrix

We define $\mathbf{S}^{(A)}(\mathbf{r})$ to be the matrix occurring in Proposition 1, i.e. we define:

$$\mathbf{S}^{(A)}(\mathbf{r}) := \begin{pmatrix} s_{\omega_1,h_1}(\mathbf{r}) & \dots & s_{\omega_1,h_{l_1}}(\mathbf{r}) \\ \vdots & & \vdots \\ s_{\omega_{n-l_0},h_1}(\mathbf{r}) & \dots & s_{\omega_{n-l_0},h_{l_1}}(\mathbf{r}) \end{pmatrix}. \tag{7}$$

Given two matrices $\mathbf{M_1}$ and $\mathbf{M_2}$, we denote by $\mathbf{M_1}|\mathbf{M_2}$ the matrix whose columns are those of $\mathbf{M_1}$ followed by those of $\mathbf{M_2}$. As a bonus of the proof of the previous proposition, we get the following:

Syndrome matrix

We define $\mathbf{S}^{(A)}(\mathbf{r})$ to be the matrix occurring in Proposition 1, i.e. we define:

$$\mathbf{S}^{(A)}(\mathbf{r}) := \begin{pmatrix} s_{\omega_1,h_1}(\mathbf{r}) & \dots & s_{\omega_1,h_{l_1}}(\mathbf{r}) \\ \vdots & & \vdots \\ s_{\omega_{n-l_0},h_1}(\mathbf{r}) & \dots & s_{\omega_{n-l_0},h_{l_1}}(\mathbf{r}) \end{pmatrix}. \tag{7}$$

Given two matrices $\mathbf{M_1}$ and $\mathbf{M_2}$, we denote by $\mathbf{M_1}|\mathbf{M_2}$ the matrix whose columns are those of $\mathbf{M_1}$ followed by those of $\mathbf{M_2}$. As a bonus of the proof of the previous proposition, we get the following:

Corollary

The rank of the matrix $\mathbf{M}_A|\mathbf{D_r}\mathbf{M}_{A-G}$ is at most $l_0 + t$, were t denotes the number of errors in $\mathbf{r}$.

Syndrome matrix

Proof:
In the proof of Proposition we defined a regular matrix H such that

$$\mathbf{H}\cdot(\mathbf{M}_A|\mathbf{D}_\mathbf{r}\mathbf{M}_{A-G}) = \left(\begin{array}{c|c} \mathbf{0} & \mathbf{S}^{(A)}(\mathbf{r}) \\ \hline \mathbf{B} & * \end{array}\right).$$

Syndrome matrix

Proof:
In the proof of Proposition we defined a regular matrix H such that

$$\mathbf{H} \cdot (\mathbf{M}_A | \mathbf{D}_\mathbf{r} \mathbf{M}_{A-G}) = \left(\begin{array}{c|c} \mathbf{0} & \mathbf{S}^{(A)}(\mathbf{r}) \\ \hline \mathbf{B} & * \end{array} \right).$$

$$\operatorname{rank}(\mathbf{M}_A | \mathbf{D}_\mathbf{r} \mathbf{M}_{A-G}) = \operatorname{rank}(\mathbf{H} \cdot (\mathbf{M}_A | \mathbf{D}_\mathbf{r} \mathbf{M}_{A-G})) = l_0 + \operatorname{rank} \mathbf{S}^{(A)}(\mathbf{r}).$$

Thus it suffices to show that $\operatorname{rank} \mathbf{S}^{(A)}(\mathbf{r}) \leq t$.

Syndrome matrix

Proof:
In the proof of Proposition we defined a regular matrix H such that

$$\mathbf{H} \cdot (\mathbf{M}_A | \mathbf{D_r M}_{A-G}) = \left(\begin{array}{c|c} \mathbf{0} & \mathbf{S}^{(A)}(\mathbf{r}) \\ \hline \mathbf{B} & * \end{array} \right).$$

$$\operatorname{rank}(\mathbf{M}_A | \mathbf{D_r M}_{A-G}) = \operatorname{rank}(\mathbf{H} \cdot (\mathbf{M}_A | \mathbf{D_r M}_{A-G})) = l_0 + \operatorname{rank} \mathbf{S}^{(A)}(\mathbf{r}).$$

Thus it suffices to show that $\operatorname{rank} \mathbf{S}^{(A)}(\mathbf{r}) \leq t$. Suppose that $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c} \in C_L(D, G)$ and $\operatorname{wt}(\mathbf{e}) = t$, then $\mathbf{S}^{(A)}(\mathbf{r}) = \mathbf{S}^{(A)}(\mathbf{e})$ and hence

$$\operatorname{rank} \mathbf{S}^{(A)}(\mathbf{r}) \leq \operatorname{rank}(\mathbf{H D_e M}_{A-G}) \leq \operatorname{rank} \mathbf{D_e} = \operatorname{wt}(\mathbf{e}) = t.$$

□

Performance of the basic algorithm

Proposition

Let $c = \mathrm{Ev}_D(f) \in C_L(D, G)$ *be a codeword and* $\mathbf{e}$ *an error-vector of weight* $t < (n - \deg G - g)/2$. *Let* $\mathbf{r} = \mathbf{c} + \mathbf{e}$, *then there exists an interpolation polynomial* $Q(y) = Q_0 + Q_1 y$ *and a divisor* A *such that*

1. $Q_0 \in L(A)$ *and* $Q_1 \in L(A - G)$,
2. $\deg A < n - t$,
3. $l(A - G) > t$,
4. $f = -Q_0/Q_1$.

Proof:

By the corollary the number of linearly independent equations in system (4) is at most $l_0 + t$.

Performance of the basic algorithm

Proof continued:
Therefore if $l(A-G) > t$ and $\deg A < n-t$, an interpolation polynomial $Q(y) = Q_0 + Q_1 y$ with the desired properties exists. If $\deg A \geq \deg G + t + g$, then $l(A-G) > t$. It is therefore enough to assume that $\deg A < n-t$ and $\deg A \geq \deg G + t + g$. A divisor A satisfying these conditions exists since $t < (n - \deg G - g)/2$. □

Performance of the basic algorithm

Proof continued:
Therefore if $l(A-G) > t$ and $\deg A < n-t$, an interpolation polynomial $Q(y) = Q_0 + Q_1 y$ with the desired properties exists. If $\deg A \geq \deg G + t + g$, then $l(A-G) > t$. It is therefore enough to assume that $\deg A < n-t$ and $\deg A \geq \deg G + t + g$. A divisor A satisfying these conditions exists since $t < (n - \deg G - g)/2$. □

Now it is time for some examples! It will illustrate all the notions and concepts introduced so far.

Example 1 In this example $\mathbb{F} = \mathbb{F}_{q^2}$, where q is a power of a prime number p. We state some general facts about the Hermitian curve χ defined over $\mathbb{F}$ by the equation

$$x_2^q + x_2 = x_1^{q+1}. \tag{8}$$

Example 1

We actually consider its projective closure, but for convenience we usually work with equation (8). First we fix some notation. Given $\alpha, \beta \in \mathbb{F}$ and a point P with $x_1(P) = \alpha$ and $x_2(P) = \beta$, we write $P = P_{\alpha\beta}$. Let $\beta_1, \ldots, \beta_q$ be all solutions to the equation $t^q + t = 0$. Then we define $T_i := P_{0\beta_i}$ for $1 \leq i \leq q$. The projective point $(0:1:0)$ we denote by T_∞. Note that the points $T_1, \ldots, T_q, T_\infty$ are exactly those points on the Hermitian curve that also lie on the line $x_1 = 0$. All these points are rational.
It is well known that the genus of H is $g = q(q-1)/2$ and that it has $q^3 + 1$ rational points. We denote the $q^3 - q$ rational points different from $T_1, \ldots, T_q, T_\infty$ by $P_1, \ldots, P_{q^3-q}$ and define

$$D := P_1 + \cdots + P_{q^3-q}.$$

Example 1

Also for any $(q+1)$-tuple $k_\infty, k_1, \ldots, k_q$ of integers we define

$$G(k_\infty, k_1, \ldots, k_q) := k_\infty T_\infty + \sum_{i=1}^{q} k_i T_i.$$

Example 1

Also for any $(q+1)$-tuple $k_\infty, k_1, \ldots, k_q$ of integers we define

$$G(k_\infty, k_1, \ldots, k_q) := k_\infty T_\infty + \sum_{i=1}^{q} k_i T_i.$$

A basis of the space $L(G(k_\infty, k_1, \ldots, k_q))$ can be described as follows: first of all, a generating set for $L(G(k_\infty, k_1, \ldots, k_q))$ is given by the set of all functions $x_1^i \prod_{j=1}^{q}(x_2 - \beta_j)^{e(i,j)}$ satisfying:

- $0 \leq i \leq q$,
- $i + (q+1)e(i,j) \geq -k_j$ for all j with $1 \leq j \leq q$,
- $iq + \sum_{j=1}^{q} e(i,j)(q+1) \leq k_\infty$.

Example 1

The resulting functions are not linearly independent in general, but this can be achieved in the following way: for each i between 0 and q and each number $d(i)$ between $-\sum_{j=1}^{q} \lfloor (k_j + i)/(q+1) \rfloor$ and $(k_\infty - iq)/(q+1)$, choose (if it exists) exactly one q-tuple $(\, e(i,1), \ldots, e(i,q)\,)$ satisfying the above conditions such that $e(i,1) + \cdots + e(i,q) = d(i)$. The corresponding functions constitute a basis.

Example 1

The resulting functions are not linearly independent in general, but this can be achieved in the following way: for each i between 0 and q and each number $d(i)$ between $-\sum_{j=1}^{q} \lfloor (k_j + i)/(q+1) \rfloor$ and $(k_\infty - iq)/(q+1)$, choose (if it exists) exactly one q-tuple $(\, e(i,1), \ldots, e(i,q)\,)$ satisfying the above conditions such that $e(i,1) + \cdots + e(i,q) = d(i)$. The corresponding functions constitute a basis. For future reference we also note that the differential dx_1 has divisor

$$(dx_1) = (q^2 - q - 2)T_\infty. \tag{9}$$

Let $S \subset \mathbb{F}_{q^2}$ and suppose that

$$D = \sum_{\alpha \in S} \sum_{\beta :\, \beta^q + \beta = \alpha^{q+1}} P_{\alpha\beta}.$$

Example 1 and Example 2

Then we have that

$$\left(\frac{dx_1}{\prod_{\alpha \in S}(x_1 - \alpha)}\right) = -D + (n + 2g - 2)T_\infty.$$

One can use this differential to show that for D as above, we obtain an isomorphism between $\Omega(-D + A)$ and $L(-A + (n + 2g - 2)T_\infty)$.

Example 1 and Example 2

Then we have that

$$\left(\frac{dx_1}{\prod_{\alpha\in S}(x_1-\alpha)}\right) = -D + (n+2g-2)T_\infty.$$

One can use this differential to show that for D as above, we obtain an isomorphism between $\Omega(-D+A)$ and $L(-A+(n+2g-2)T_\infty)$.

Example 2 In this example consider the Hermitian curve for $q=4$ and choose the divisor $G = T_1 + 2T_2 + 3T_3 + 4T_4 + 13T_\infty$. We write $\mathbb{F}_{16}$ as $\mathbb{F}_2[\gamma]$, where $\gamma^4 = \gamma + 1$. All solutions of $t^4 + t = 0$ are then given by $\beta_1 = 0$, $\beta_2 = 1$, $\beta_3 = \gamma^5$, and $\beta_4 = \gamma^{10}$.

Example 2

A basis for $L(G)$ is given by

- x_2^α, with $0 \leq \alpha \leq 2$,
- $x_1 x_2^\alpha/(x_2+\gamma^{10})$, with $0 \leq \alpha \leq 2$,
- $x_1^2 x_2^\alpha/(x_2^2+x_2+1)$, with $0 \leq \alpha \leq 3$,
- $x_1^3 x_2^\alpha/(x_2^3+1)$, with $0 \leq \alpha \leq 3$, and
- $x_1^4 x_2^\alpha/(x_2^4+x_2)$, with $0 \leq \alpha \leq 3$.

¡pause Now let D be the sum of all 60 rational points not in $\mathrm{supp}\, G$. We order the points by writing their coordinates as a power of γ and then ordering theses two exponents lexicographically. In this way we get $P_1 = (1, \gamma), \ldots, P_{60} = (\gamma^{14}, \gamma^{14})$.

Example 2

The code $C_L(D, G)$ is an $[60, 18, \geq 37]$ code and the basic algorithm can correct $t = 15$ errors. Now we choose $A = G + 21T_\infty$, since then $\deg A = 44 < 60 - 15$ and $l(A - G) = l(21T_\infty) = 16 > 15$.

Example 2

The code $C_L(D, G)$ is an $[60, 18, \geq 37]$ code and the basic algorithm can correct $t = 15$ errors. Now we choose $A = G + 21T_\infty$, since then $\deg A = 44 < 60 - 15$ and $l(A - G) = l(21T_\infty) = 16 > 15$. To write down system (5), we need, according to Proposition 1, to calculate a basis for the space $L(A - G)$ and differentials $\omega_1, \ldots, \omega_{21}$ such that their images under the residue map form a basis of the code $C_\Omega(D, A)$. In this case the last part amounts to calculating a basis for $\Omega(-D + A)$.

Example 2

The code $C_L(D, G)$ is an $[60, 18, \geq 37]$ code and the basic algorithm can correct $t = 15$ errors. Now we choose $A = G + 21T_\infty$, since then $\deg A = 44 < 60 - 15$ and $l(A - G) = l(21T_\infty) = 16 > 15$. To write down system (5), we need, according to Proposition 1, to calculate a basis for the space $L(A - G)$ and differentials $\omega_1, \ldots, \omega_{21}$ such that their images under the residue map form a basis of the code $C_\Omega(D, A)$. In this case the last part amounts to calculating a basis for $\Omega(-D + A)$. Using the differential form $\omega := (x_1^{15} + 1)^{-1} dx_1$, we see that the spaces $L(-A + 70T_\infty)$ and $\Omega(-D + A)$ are isomorphic via $f \mapsto f\omega$.

Example 2

A basis for $L(A - G)$ is given by:

- x_2^α, with $0 \leq \alpha \leq 4$,
- $x_1 x_2^\alpha$, with $0 \leq \alpha \leq 3$,
- $x_1^2 x_2^\alpha$, with $0 \leq \alpha \leq 2$,
- $x_1^3 x_2^\alpha$, with $0 \leq \alpha \leq 1$, and
- $x_1^4 x_2^\alpha$, with $0 \leq \alpha \leq 1$.

Example 2

A basis for $L(A - G)$ is given by:

- x_2^α, with $0 \leq \alpha \leq 4$,
- $x_1 x_2^\alpha$, with $0 \leq \alpha \leq 3$,
- $x_1^2 x_2^\alpha$, with $0 \leq \alpha \leq 2$,
- $x_1^3 x_2^\alpha$, with $0 \leq \alpha \leq 1$, and
- $x_1^4 x_2^\alpha$, with $0 \leq \alpha \leq 1$.

We order this basis with respect to the pole-order in T_∞, so that $h_1 = 1$, $h_2 = x_1$,$h_3 = x_2$, ..., $h_{15} = x_2^4$, $h_{16} = x_1^4 x_2$. A basis for $\Omega(-D + A)$ is given by:

- $(x_2^4 + x_2)x_2^\alpha \omega$, with $0 \leq \alpha \leq 3$,
- $x_1(x_2^3 + 1)x_2^\alpha \omega$, with $0 \leq \alpha \leq 3$,
- $x_1^2(x_2^2 + x_2 + 1)x_2^\alpha \omega$, with $0 \leq \alpha \leq 3$,
- $x_1^3(x_2 + \gamma^{10})x_2^\alpha \omega$, with $0 \leq \alpha \leq 3$, and
- $x_1^4 x_2^\alpha \omega$, with $0 \leq \alpha \leq 4$.

Example 2

Again we order this basis with respect to the pole-order in T_∞. We then get $\omega_1 = x_1^4\omega$, $\omega_2 = x_1^3(x_2 - \gamma^{10})\omega$, $\dots$, $\omega_{20} = (x_2^4 + x_2)x_2^3\omega$, $\omega_{21} = x_1^4 x_2^4\omega$.

Example 2

Again we order this basis with respect to the pole-order in T_∞. We then get $\omega_1 = x_1^4\omega$, $\omega_2 = x_1^3(x_2 - \gamma^{10})\omega$, $\dots$, $\omega_{20} = (x_2^4 + x_2)x_2^3\omega$, $\omega_{21} = x_1^4 x_2^4\omega$. Now we will show an example of error-correction using the basic algorithm. Suppose that the sent codeword is $\mathbf{c} = \mathrm{Ev}_D\left(x_2^2 + x_1^4 x_2^3/(x_2^4 + x_2)\right)$ and that the error-vector $\mathbf{e} = (e_1, \dots, e_{60})$ is given by $e_4 = 1$, $e_8 = \gamma$, $e_9 = \gamma^3$, $e_{16} = \gamma^7$, $e_{18} = \gamma^{11}$, $e_{25} = 1$, $e_{31} = \gamma$, $e_{37} = \gamma^6$, $e_{39} = \gamma^{10}$, $e_{42} = \gamma$, $e_{47} = 1$, $e_{52} = \gamma^{12}$, $e_{55} = \gamma^8$, $e_{58} = 1$, $e_{60} = \gamma^3$, and $e_i = 0$ for all other values of i.

Example 2

Again we order this basis with respect to the pole-order in T_∞. We then get $\omega_1 = x_1^4\omega$, $\omega_2 = x_1^3(x_2 - \gamma^{10})\omega$, $\ldots$, $\omega_{20} = (x_2^4 + x_2)x_2^3\omega$, $\omega_{21} = x_1^4x_2^4\omega$. Now we will show an example of error-correction using the basic algorithm. Suppose that the sent codeword is $\mathbf{c} = \mathrm{Ev}_D\left(x_2^2 + x_1^4x_2^3/(x_2^4 + x_2)\right)$ and that the error-vector $\mathbf{e} = (e_1, \ldots, e_{60})$ is given by $e_4 = 1$, $e_8 = \gamma$, $e_9 = \gamma^3$, $e_{16} = \gamma^7$, $e_{18} = \gamma^{11}$, $e_{25} = 1$, $e_{31} = \gamma$, $e_{37} = \gamma^6$, $e_{39} = \gamma^{10}$, $e_{42} = \gamma$, $e_{47} = 1$, $e_{52} = \gamma^{12}$, $e_{55} = \gamma^8$, $e_{58} = 1$, $e_{60} = \gamma^3$, and $e_i = 0$ for all other values of i. The matrix $\mathbf{S}^{(A)}(\mathbf{c} + \mathbf{e})$, which is independent of the sent codeword $\mathbf{c}$, is the following:

Example 2 - The syndrome matrix

$$
\begin{pmatrix}
\gamma^{6} & \gamma^{5} & \gamma^{14} & \gamma^{11} & \gamma^{6} & \gamma & \gamma^{13} & \gamma & \gamma^{6} & \gamma^{2} & 0 & \gamma^{12} & 0 & \gamma^{2} & \gamma^{6} & \gamma^{9} \\
\gamma^{9} & \gamma^{7} & \gamma^{5} & \gamma^{13} & \gamma^{3} & \gamma^{11} & \gamma^{11} & \gamma^{11} & \gamma^{9} & \gamma^{7} & \gamma^{9} & \gamma^{11} & \gamma^{5} & \gamma^{4} & \gamma^{6} & \gamma^{5} \\
\gamma^{3} & \gamma^{12} & \gamma^{10} & \gamma^{10} & \gamma^{14} & \gamma^{9} & \gamma^{5} & \gamma^{12} & \gamma^{14} & \gamma^{8} & \gamma^{6} & \gamma^{2} & \gamma^{9} & \gamma^{4} & \gamma^{3} & \gamma \\
1 & \gamma^{12} & \gamma^{11} & \gamma^{5} & \gamma^{13} & \gamma & \gamma^{3} & 0 & \gamma^{12} & 0 & \gamma & \gamma^{8} & \gamma^{9} & \gamma^{13} & 0 & 1 \\
\gamma^{5} & \gamma^{11} & \gamma^{6} & \gamma^{13} & \gamma & \gamma^{6} & 0 & \gamma^{12} & 0 & \gamma^{2} & \gamma^{8} & \gamma^{9} & \gamma^{13} & 0 & \gamma^{13} & \gamma^{10} \\
\gamma^{14} & \gamma^{6} & \gamma & \gamma & \gamma^{6} & \gamma^{2} & \gamma^{12} & 0 & \gamma^{2} & \gamma^{6} & \gamma^{9} & \gamma^{13} & 0 & \gamma^{13} & \gamma^{8} & \gamma^{7} \\
\gamma^{5} & \gamma^{3} & \gamma^{11} & \gamma^{11} & \gamma^{9} & \gamma^{7} & \gamma^{11} & \gamma^{5} & \gamma^{4} & \gamma^{6} & \gamma^{5} & 0 & \gamma & \gamma^{10} & \gamma^{5} & \gamma^{7} \\
\gamma^{10} & \gamma^{14} & \gamma^{9} & \gamma^{12} & \gamma^{14} & \gamma^{8} & \gamma^{2} & \gamma^{9} & \gamma^{4} & \gamma^{3} & \gamma & \gamma^{8} & \gamma^{13} & \gamma^{3} & \gamma^{5} & \gamma^{5} \\
\gamma^{11} & \gamma^{13} & \gamma & 0 & \gamma^{12} & 0 & \gamma^{8} & \gamma^{9} & \gamma^{13} & 0 & 1 & \gamma^{10} & \gamma^{7} & \gamma^{11} & \gamma^{5} & \gamma^{11} \\
\gamma^{6} & \gamma & \gamma^{6} & \gamma^{12} & 0 & \gamma^{2} & \gamma^{9} & \gamma^{13} & 0 & \gamma^{13} & \gamma^{10} & \gamma^{7} & \gamma^{11} & \gamma^{5} & \gamma & \gamma^{12} \\
\gamma & \gamma^{6} & \gamma^{2} & 0 & \gamma^{2} & \gamma^{6} & \gamma^{13} & 0 & \gamma^{13} & \gamma^{8} & \gamma^{7} & \gamma^{11} & \gamma^{5} & \gamma & \gamma^{7} & \gamma^{4} \\
\gamma^{11} & \gamma^{9} & \gamma^{7} & \gamma^{5} & \gamma^{4} & \gamma^{6} & 0 & \gamma & \gamma^{10} & \gamma^{5} & \gamma^{7} & \gamma^{5} & \gamma^{10} & \gamma^{4} & \gamma^{6} & \gamma^{14} \\
\gamma^{9} & \gamma^{14} & \gamma^{8} & \gamma^{9} & \gamma^{4} & \gamma^{3} & \gamma^{8} & \gamma^{13} & \gamma^{3} & \gamma^{5} & \gamma^{5} & \gamma^{7} & \gamma & \gamma^{7} & \gamma^{4} & \gamma^{13} \\
\gamma & \gamma^{12} & 0 & \gamma^{9} & \gamma^{13} & 0 & \gamma^{10} & \gamma^{7} & \gamma^{11} & \gamma^{5} & \gamma^{11} & \gamma^{12} & \gamma^{4} & \gamma^{8} & \gamma^{7} & \gamma^{11} \\
\gamma^{6} & 0 & \gamma^{2} & \gamma^{13} & 0 & \gamma^{13} & \gamma^{7} & \gamma^{11} & \gamma^{5} & \gamma & \gamma^{12} & \gamma^{4} & \gamma^{8} & \gamma^{7} & \gamma^{9} & \gamma^{13} \\
\gamma^{2} & \gamma^{2} & \gamma^{6} & 0 & \gamma^{13} & \gamma^{8} & \gamma^{11} & \gamma^{5} & \gamma & \gamma^{7} & \gamma^{4} & \gamma^{8} & \gamma^{7} & \gamma^{9} & 1 & \gamma^{8} \\
\gamma^{7} & \gamma^{4} & \gamma^{6} & \gamma & \gamma^{10} & \gamma^{5} & \gamma^{5} & \gamma^{10} & \gamma^{4} & \gamma^{6} & \gamma^{14} & \gamma^{9} & \gamma^{2} & \gamma^{8} & 0 & \gamma^{2} \\
\gamma^{8} & \gamma^{4} & \gamma^{3} & \gamma^{13} & \gamma^{3} & \gamma^{5} & \gamma^{7} & \gamma & \gamma^{7} & \gamma^{4} & \gamma^{13} & \gamma^{8} & \gamma^{12} & \gamma^{11} & \gamma^{12} & \gamma^{14} \\
0 & \gamma^{13} & 0 & \gamma^{7} & \gamma^{11} & \gamma^{5} & \gamma^{12} & \gamma^{4} & \gamma^{8} & \gamma^{7} & \gamma^{11} & \gamma^{13} & \gamma^{8} & \gamma^{6} & \gamma^{2} & \gamma^{6} \\
\gamma^{2} & 0 & \gamma^{13} & \gamma^{11} & \gamma^{5} & \gamma & \gamma^{4} & \gamma^{8} & \gamma^{7} & \gamma^{9} & \gamma^{13} & \gamma^{8} & \gamma^{6} & \gamma^{2} & 1 & 1 \\
\gamma^{6} & \gamma^{13} & \gamma^{8} & \gamma^{5} & \gamma & \gamma^{7} & \gamma^{8} & \gamma^{7} & \gamma^{9} & 1 & \gamma^{8} & \gamma^{6} & \gamma^{2} & 1 & \gamma^{2} & \gamma^{6}
\end{pmatrix}
$$

Example 2

One can check that the kernel of this matrix is one-dimensional. A corresponding error-locator is:

$$Q_1 = \gamma^{12}h_2 + h_3 + \gamma^2 h_4 + \gamma^2 h_5 + \gamma^4 h_6 + \gamma^{13} h_7 + \gamma^6 h_8 + \gamma^7 h_9 + \gamma^4 h_{10} + \gamma^3 h_{11} + \gamma^7 h_{12} + \gamma^6 h_{13} + \gamma^{11} h_{14} + \gamma^8 h_{15}.$$

Example 2

One can check that the kernel of this matrix is one-dimensional. A corresponding error-locator is:

$$Q_1 = \gamma^{12} h_2 + h_3 + \gamma^2 h_4 + \gamma^2 h_5 + \gamma^4 h_6 + \gamma^{13} h_7 + \gamma^6 h_8 + \\ \gamma^7 h_9 + \gamma^4 h_{10} + \gamma^3 h_{11} + \gamma^7 h_{12} + \gamma^6 h_{13} + \gamma^{11} h_{14} + \gamma^8 h_{15}.$$

The error-positions i can be found by computing the zeroes P_i of this polynomial. In this case we find that the 15 error-positions are contained in the set $\{4, 8, 9, 12, 16, 18, 19, 21, 25, 31, 37, 39, 42, 47, 48, 52, 55, 58, 60\}$.

Example 2

Now that the variables $\mathbf{q}_1 = (q_{11}, \ldots, q_{1l_1})$ are known, we can substitute their values into system (6). In that way we obtain a system of 39 equations in the 39 variables $\mathbf{q}_0 = (q_{01}, \ldots, q_{0l_0})$.

Example 2

Now that the variables $\mathbf{q}_1 = (q_{11}, \ldots, q_{1l_1})$ are known, we can substitute their values into system (6). In that way we obtain a system of 39 equations in the 39 variables $\mathbf{q}_0 = (q_{01}, \ldots, q_{0l_0})$. To find these equations we need to choose, as in Lemma 5, differentials $\omega_1, \ldots, \omega_{60}$ such that their images under the map Res_D form a basis of $\mathbb{F}_{16}^{60}$. The first 21 are simply the differentials defined above as the basis for $\Omega(-D+A)$. The remaining 39 we choose from $\Omega(-D+A-45T_\infty)$. We can actually choose them in the following way

- $(x_2^4 + x_2)x_2^\alpha\omega$, with $4 \le \alpha \le 11$,
- $x_1(x_2^3 + 1)x_2^\alpha\omega$, with $4 \le \alpha \le 11$,
- $x_1^2(x_2^2 + x_2 + 1)x_2^\alpha\omega$, with $4 \le \alpha \le 11$,
- $x_1^3(x_2 + \gamma^{10})x_2^\alpha\omega$, with $4 \le \alpha \le 11$, and
- $x_1^4x_2^\alpha\omega$, with $5 \le \alpha \le 11$.

Example 2

Like for the given basis for $\Omega(-D+A)$, we order this basis by increasing pole order at T_∞. Then we get $\omega_{22} = x_1^3(x_2+\gamma^{10})x_2^4\omega, \ldots, \omega_{60} = (x_2^4+x_2)x_2^{11}$. We can now calculate the 60×60 matrix $\mathbf{H}$ as well as the vector $\mathbf{v} := \mathbf{H}\mathbf{D}_\mathbf{r}\mathbf{M}_{A-G}\mathbf{q}_1$. The first 21 coordinates of $\mathbf{v}$ are 0, since $\mathbf{q}_1$ is in the kernel of $\mathbf{S}^{(A)}(\mathbf{r})$. The remaining 39 coordinates of this vector $(v_{22}, \ldots, v_{60})$ are given by:

$$(0,0,0,\gamma^8,\gamma^7,\gamma,\gamma^{10},\gamma^4,\gamma^7,\gamma^3,\gamma^{14},\gamma^5,\gamma^{13},\gamma^4,\gamma^{10},\gamma^5,\gamma,0,\gamma^2,\gamma^8,$$
$$\gamma^{13},\gamma,0,\gamma^4,\gamma^3,\gamma,\gamma,0,\gamma^4,\gamma^{10},\gamma^5,\gamma,0,1,\gamma^{11},\gamma^{12},\gamma^8,\gamma^4,\gamma^3).$$

Example 2

We now choose the following basis for $L(A)$:

- x_2^α, with $0 \leq \alpha \leq 6$,
- $x_1 x_2^\alpha/(x_2 + \gamma^{10})$, with $0 \leq \alpha \leq 7$,
- $x_1^2 x_2^\alpha/(x_2^2 + x_2 + 1)$, with $0 \leq \alpha \leq 7$,
- $x_1^3 x_2^\alpha/(x_2^3 + 1)$, with $0 \leq \alpha \leq 7$, and
- $x_1^4 x_2^\alpha/(x_2^4 + x_2)$, with $0 \leq \alpha \leq 7$,

and order it with increasing pole order in T_∞. Then $g_1 = x_1^4/(x_2^4 + x_2)$, $g_2 = x_1^3/(x_2^3 + 1), \ldots, g_{39} = x_1 x_2^7/(x_2 + \gamma^{10})$. We can then calculate the matrix $\mathbf{B}$ from the proof of Proposition 1.

Example 2

We now choose the following basis for $L(A)$:

- x_2^{α}, with $0 \leq \alpha \leq 6$,
- $x_1 x_2^{\alpha}/(x_2 + \gamma^{10})$, with $0 \leq \alpha \leq 7$,
- $x_1^2 x_2^{\alpha}/(x_2^2 + x_2 + 1)$, with $0 \leq \alpha \leq 7$,
- $x_1^3 x_2^{\alpha}/(x_2^3 + 1)$, with $0 \leq \alpha \leq 7$, and
- $x_1^4 x_2^{\alpha}/(x_2^4 + x_2)$, with $0 \leq \alpha \leq 7$,

and order it with increasing pole order in T_∞. Then $g_1 = x_1^4/(x_2^4 + x_2)$, $g_2 = x_1^3/(x_2^3 + 1), \ldots, g_{39} = x_1 x_2^7/(x_2 + \gamma^{10})$. We can then calculate the matrix $\mathbf{B}$ from the proof of Proposition 1. By the way we have chosen and ordered the differentials and functions, we obtain more structure than was indicated in Proposition 1. In this case we obtain that

$$\mathbf{B}_{ij} = \begin{cases} 1 & \text{if } i+j = 40 \text{ or } i+j = 55, \\ 0 & \text{otherwise.} \end{cases}$$

Example 2

This means that is straightforward to calculate Q_0 now and we obtain

$$\begin{aligned} Q_0 \;=\; & \gamma^{13}g_{12}+\gamma^{2}g_{13}+\gamma^{7}g_{14}+\gamma^{3}g_{16}+\gamma^{4}g_{17}+\gamma^{8}g_{18}+\gamma^{12}g_{19} \\ & +\gamma^{11}g_{20}+g_{21}+\gamma g_{23}+\gamma^{5}g_{24}+\gamma^{10}g_{25}+\gamma^{4}g_{26}+\gamma^{13}g_{27} \\ & +\gamma^{5}g_{28}+\gamma^{14}g_{29}+\gamma^{3}g_{30}+\gamma^{7}g_{31}+\gamma^{4}g_{32}+\gamma^{10}g_{33} \\ & +\gamma g_{34}+\gamma^{7}g_{35}+\gamma^{8}g_{36}. \end{aligned}$$

Note that $Q_0/Q_1 = x_2^2 + x_1^4x_2^3/(x_2^4+x_2)$.

Contents

The generalized order bound

- The Goppa–bound for $C_L(D, G)$ is $d \geq n - \deg G$.
- The Goppa–bound for $C_\Omega(D, G)$ is $d \geq \deg G - 2g + 2$.

The generalized order bound

- The Goppa–bound for $C_L(D, G)$ is $d \geq n - \deg G$.
- The Goppa–bound for $C_\Omega(D, G)$ is $d \geq \deg G - 2g + 2$.
- If $\deg G \leq 2g - 2$ the bound $d \geq \deg G - 2g + 2$ is trivial, while if $\deg G \geq n$, the bound $d \geq n - \deg G$ lower bound is trivial.

The generalized order bound

- The Goppa–bound for $C_L(D, G)$ is $d \geq n - \deg G$.
- The Goppa–bound for $C_\Omega(D, G)$ is $d \geq \deg G - 2g + 2$.
- If $\deg G \leq 2g - 2$ the bound $d \geq \deg G - 2g + 2$ is trivial, while if $\deg G \geq n$, the bound $d \geq n - \deg G$ lower bound is trivial.
- We will see that there exist a bound (the generalized order bound) that improves the Goppa-bounds in the mentioned cases, but sometimes also if $2g - 2 < \deg G < n$.

Weierstrass semigroups

Let $T \notin \operatorname{supp} D$ be a rational point. We then define the ring

$$R(T) := \bigcup_{i \geq 0} L(iT). \qquad (10)$$

There is a natural mapping ρ_T from $R(T) \backslash \{0\}$ to $\mathbb{N} = \{0, 1, 2, \ldots\}$, namely

$$f \mapsto -v_T(f). \qquad (11)$$

The image $H(T)$ of this map is the so-called Weierstrass semigroup of T:

$$H(T) := \rho_T(R(T) \backslash \{0\}). \qquad (12)$$

Weierstrass semigroups

Let $T \notin \operatorname{supp} D$ be a rational point. We then define the ring

$$R(T) := \bigcup_{i \geq 0} L(iT). \tag{10}$$

There is a natural mapping ρ_T from $R(T) \setminus \{0\}$ to $\mathbb{N} = \{0, 1, 2, \ldots\}$, namely

$$f \mapsto -v_T(f). \tag{11}$$

The image $H(T)$ of this map is the so-called Weierstrass semigroup of T:

$$H(T) := \rho_T(R(T) \setminus \{0\}). \tag{12}$$

We will define a certain $R(T)$-modules called *order modules* that will be used to obtain lower bounds on the minimum distance of AG–codes.

Order modules

Definition

An *order module* $\mathcal{M}$ for $R(T)$ is a pair (M, φ), where M is an $R(T)$-module and φ a surjective $\mathbb{F}$-linear map $\varphi : M \to \mathbb{F}^n$ s.t.:

1. $M = \bigcup_{i \in \mathbb{Z}} M_i$, with $M_i \subset M$ vector spaces such that for all integers $i \leq j$ we have that $M_i \subset M_j$,

Order modules

Definition

An *order module* $\mathcal{M}$ for $R(T)$ is a pair (M, φ), where M is an $R(T)$-module and φ a surjective $\mathbb{F}$-linear map $\varphi : M \to \mathbb{F}^n$ s.t.:

1. $M = \bigcup_{i \in \mathbb{Z}} M_i$, with $M_i \subset M$ vector spaces such that for all integers $i \leq j$ we have that $M_i \subset M_j$,
2. There exists an integer a such that $M_i = \{0\}$ for all $i < a$,

Order modules

Definition

An *order module* $\mathcal{M}$ for $R(T)$ is a pair (M, φ), where M is an $R(T)$-module and φ a surjective $\mathbb{F}$-linear map $\varphi : M \to \mathbb{F}^n$ s.t.:

1. $M = \bigcup_{i \in \mathbb{Z}} M_i$, with $M_i \subset M$ vector spaces such that for all integers $i \leq j$ we have that $M_i \subset M_j$,
2. There exists an integer a such that $M_i = \{0\}$ for all $i < a$,
3. For any integers i and j, we have that $L(iT)M_j \subset M_{i+j}$,

Order modules

Definition

An *order module* $\mathcal{M}$ for $R(T)$ is a pair (M, φ), where M is an $R(T)$-module and φ a surjective $\mathbb{F}$-linear map $\varphi : M \to \mathbb{F}^n$ s.t.:

1. $M = \bigcup_{i \in \mathbb{Z}} M_i$, with $M_i \subset M$ vector spaces such that for all integers $i \leq j$ we have that $M_i \subset M_j$,
2. There exists an integer a such that $M_i = \{0\}$ for all $i < a$,
3. For any integers i and j, we have that $L(iT)M_j \subset M_{i+j}$,
4. For $f \in R(T)$, $m \in M$ it holds $\varphi(fm) = \mathrm{Ev}_D(f) * \varphi(m)$. Here $*$ is coordinate-wise product on $\mathbb{F}^n$,

Order modules

Definition

An *order module* $\mathcal{M}$ for $R(T)$ is a pair (M, φ), where M is an $R(T)$-module and φ a surjective $\mathbb{F}$-linear map $\varphi : M \to \mathbb{F}^n$ s.t.:

1. $M = \bigcup_{i \in \mathbb{Z}} M_i$, with $M_i \subset M$ vector spaces such that for all integers $i \leq j$ we have that $M_i \subset M_j$,
2. There exists an integer a such that $M_i = \{0\}$ for all $i < a$,
3. For any integers i and j, we have that $L(iT)M_j \subset M_{i+j}$,
4. For $f \in R(T)$, $m \in M$ it holds $\varphi(fm) = \mathrm{Ev}_D(f) * \varphi(m)$. Here $*$ is coordinate-wise product on $\mathbb{F}^n$,
5. For $m \in M_i \backslash M_{i-1}$ and $f \in R(T)$ satisfying $\rho_T(f) = j$, we have that $fm \in M_{i+j} \backslash M_{i+j-1}$,

Order modules

Definition

An *order module* $\mathcal{M}$ for $R(T)$ is a pair (M, φ), where M is an $R(T)$-module and φ a surjective $\mathbb{F}$-linear map $\varphi : M \to \mathbb{F}^n$ s.t.:

1. $M = \bigcup_{i \in \mathbb{Z}} M_i$, with $M_i \subset M$ vector spaces such that for all integers $i \leq j$ we have that $M_i \subset M_j$,
2. There exists an integer a such that $M_i = \{0\}$ for all $i < a$,
3. For any integers i and j, we have that $L(iT)M_j \subset M_{i+j}$,
4. For $f \in R(T)$, $m \in M$ it holds $\varphi(fm) = \mathrm{Ev}_D(f) * \varphi(m)$. Here $*$ is coordinate-wise product on $\mathbb{F}^n$,
5. For $m \in M_i \backslash M_{i-1}$ and $f \in R(T)$ satisfying $\rho_T(f) = j$, we have that $fm \in M_{i+j} \backslash M_{i+j-1}$,
6. For all i, we have that $M_i = M_{i-1}$ or $\dim M_i = \dim M_{i-1} + 1$.

Order modules

Remark

An analogue of the map ρ_T can be defined on $\mathcal{M}$ as follows:

$$\rho_{T,\mathcal{M}} : M\backslash\{0\} \to \mathbb{Z}, \quad m \mapsto \min\{i \mid m \in M_i\}. \tag{13}$$

Item (5) of the definition then reads

(5a) *For $f \in R(T)\backslash\{0\}$, $m \in M\backslash\{0\}$ we have that $\rho_{T,M}(fm) = \rho_T(f) + \rho_{T,M}(m)$.*

Order modules

Remark

An analogue of the map ρ_T can be defined on $\mathcal{M}$ as follows:

$$\rho_{T,\mathcal{M}} : M\backslash\{0\} \to \mathbb{Z}, \quad m \mapsto \min\{i \mid m \in M_i\}. \tag{13}$$

Item (5) of the definition then reads

(5a) *For $f \in R(T)\backslash\{0\}$, $m \in M\backslash\{0\}$ we have that* $\rho_{T,M}(fm) = \rho_T(f) + \rho_{T,M}(m)$.

The linear subspaces $\varphi(M_i) \subset \mathbb{F}^n$ are interpreted as codes. Examples of order modules are:

$$\mathcal{M}_L(D, G, T) := (\cup_{i\in\mathbb{Z}} L(G + iT), \mathrm{Ev}_D\,) \tag{14}$$

$$\mathcal{M}_\Omega(D, G, T) := (\cup_{i\in\mathbb{Z}} \Omega(-D + G - iT), \mathrm{Res}_D\,). \tag{15}$$

Order modules

In the first case, we have that $\rho_{T,\mathcal{M}}(m) = -v_T(m) - v_T(G)$, while the corresponding codes are the codes $C_L(D, G + iT)$. In the second example we have that $\rho_{T,\mathcal{M}}(m) = -v_T(m) + v_T(G)$, while we now obtain the codes $C_\Omega(D, G - iT)$.

Order modules

In the first case, we have that $\rho_{T,\mathcal{M}}(m) = -v_T(m) - v_T(G)$, while the corresponding codes are the codes $C_L(D, G + iT)$. In the second example we have that $\rho_{T,\mathcal{M}}(m) = -v_T(m) + v_T(G)$, while we now obtain the codes $C_\Omega(D, G - iT)$.

Remark

The codes coming from $\mathcal{M}_\Omega(D, G, T)$ are the same as those from $\mathcal{M}_L(D, K + D - G, T)$, where $K = (\omega)$ is the divisor of a differential ω that has poles of order one and residues equal to one in all points of $\operatorname{supp} D$. If one wishes, we can therefore reduce computations in the module $\mathcal{M}_\Omega(D, G, T)$ to ones in $\mathcal{M}_L(D, K + D - G, T)$.

Generalized Weierstrass semigroups and gaps

The analogue of the set $H(T)$ for an order module $\mathcal{M} = (M, \varphi)$ is:

$$H(T, \mathcal{M}) := \rho_{T,\mathcal{M}}(M \setminus \{0\}). \tag{16}$$

Note that this set is not a semigroup in general, but it does have the property that $i \in H(T, \mathcal{M})$ implies that $i + H(T) \subset H(T, \mathcal{M})$.

Generalized Weierstrass semigroups and gaps

The analogue of the set $H(T)$ for an order module $\mathcal{M} = (M, \varphi)$ is:

$$H(T, \mathcal{M}) := \rho_{T,\mathcal{M}}(M \setminus \{0\}). \tag{16}$$

Note that this set is not a semigroup in general, but it does have the property that $i \in H(T, \mathcal{M})$ implies that $i + H(T) \subset H(T, \mathcal{M})$. An element from $\mathbb{N} \setminus H(T)$ is called a gap of the semigroup $H(T)$. It is well known that the number of gaps equals the genus g of the curve. We will define the analogue concepts for $H(T, \mathcal{M})$.

Generalized Weierstrass semigroups and gaps

The analogue of the set $H(T)$ for an order module $\mathcal{M} = (M, \varphi)$ is:

$$H(T, \mathcal{M}) := \rho_{T,\mathcal{M}}(M \backslash \{0\}). \tag{16}$$

Note that this set is not a semigroup in general, but it does have the property that $i \in H(T, \mathcal{M})$ implies that $i + H(T) \subset H(T, \mathcal{M})$. An element from $\mathbb{N} \backslash H(T)$ is called a gap of the semigroup $H(T)$. It is well known that the number of gaps equals the genus g of the curve. We will define the analogue concepts for $H(T, \mathcal{M})$.

Definition

Let $a = \min H(T, \mathcal{M})$. The set $\mathbb{Z}_{\geq a} \backslash H(T, \mathcal{M})$ is called the set of *gaps* of $H(T, \mathcal{M})$. We denote the number of gaps by $g(\mathcal{M})$.

Definitions for the generalized order bound

Since $a + H(T) \subset H(T, \mathcal{M})$, we always have $g(\mathcal{M}) \leq g$. Using Riemann-Roch's theorem, get

- $a = -\deg G + g - g(\mathcal{M})$ if $\mathcal{M} = \mathcal{M}_L(D, G, T)$.
- $a = -n + \deg G - g - g(\mathcal{M}) + 2$ if $\mathcal{M} = \mathcal{M}_\Omega(D, G, T)$.

Definitions for the generalized order bound

Since $a + H(T) \subset H(T, \mathcal{M})$, we always have $g(\mathcal{M}) \leq g$. Using Riemann-Roch's theorem, get

- $a = -\deg G + g - g(\mathcal{M})$ if $\mathcal{M} = \mathcal{M}_L(D, G, T)$.
- $a = -n + \deg G - g - g(\mathcal{M}) + 2$ if $\mathcal{M} = \mathcal{M}_\Omega(D, G, T)$.

To formulate the generalized order bound we introduce:

$$N(T, \mathcal{M}, i) := \{(i_1, i_2) \mid i_1 \in H(T); i_2 \in H(T, \mathcal{M}); i_1 + i_2 = i + 1\}$$

$$\nu(T, \mathcal{M}, i) := \# N(T, \mathcal{M}, i).$$

Definitions for the generalized order bound

Since $a + H(T) \subset H(T, \mathcal{M})$, we always have $g(\mathcal{M}) \leq g$. Using Riemann-Roch's theorem, get

- $a = -\deg G + g - g(\mathcal{M})$ if $\mathcal{M} = \mathcal{M}_L(D, G, T)$.
- $a = -n + \deg G - g - g(\mathcal{M}) + 2$ if $\mathcal{M} = \mathcal{M}_\Omega(D, G, T)$.

To formulate the generalized order bound we introduce:

$$N(T, \mathcal{M}, i) := \{(i_1, i_2) \,|\, i_1 \in H(T); i_2 \in H(T, \mathcal{M}); i_1 + i_2 = i + 1\}$$

$$\nu(T, \mathcal{M}, i) := \#N(T, \mathcal{M}, i).$$

Lemma

Let $p_T(t) := \sum_{i_1 \in H(T)} t^{i_1}$ and $p_{T,\mathcal{M}}(t) := \sum_{i_2 \in H(T,\mathcal{M})} t^{i_2}$. Then $\nu(T, \mathcal{M}, i)$ is the coefficient of t^{i+1} in $p_T(t) p_{T,\mathcal{M}}(t)$.

This is by definition of $\nu(T, \mathcal{M}, i)$.

Counting with series

Using the series interpretation we can get a lower bound on $\nu(T, \mathcal{M}, i)$.

Lemma

Let $\mathcal{M}$ be an order module and let $a = \min H(T, \mathcal{M})$. Then $\nu(T, \mathcal{M}, i) \geq i - a + 2 - g - g(\mathcal{M})$.

Counting with series

Using the series interpretation we can get a lower bound on $\nu(T, \mathcal{M}, i)$.

Lemma

Let $\mathcal{M}$ be an order module and let $a = \min H(T, \mathcal{M})$. Then $\nu(T, \mathcal{M}, i) \geq i - a + 2 - g - g(\mathcal{M})$.

Proof:

We can choose polynomials $q_T(t)$ and $q_{T,\mathcal{M}}(t)$ such that the following identities of Laurent series hold:

$$p_T(t) + q_T(t) = \frac{1}{1-t}, \quad p_{T,\mathcal{M}}(t) + q_{T,\mathcal{M}}(t) = \frac{t^a}{1-t}.$$

Counting with series

$q_T(t)$ is the sum of precisely g monomials, and $q_{T,\mathcal{M}}(t)$ of $g(\mathcal{M})$ monomials. These monomials all have coefficient 1. We get

$$p_T(t)p_{T,\mathcal{M}}(t) = t^a\frac{1}{(1-t)^2} - \frac{t^a q_T(t) + q_{T,\mathcal{M}}(t)}{1-t} + q_T(t)q_{T,\mathcal{M}}(t).$$

Counting with series

$q_T(t)$ is the sum of precisely g monomials, and $q_{T,\mathcal{M}}(t)$ of $g(\mathcal{M})$ monomials. These monomials all have coefficient 1. We get

$$p_T(t)p_{T,\mathcal{M}}(t) = t^a \frac{1}{(1-t)^2} - \frac{t^a q_T(t) + q_{T,\mathcal{M}}(t)}{1-t} + q_T(t)q_{T,\mathcal{M}}(t).$$

Considering this as a Laurent series in t, we can compute the coefficient of t^{i+1}. The term $t^a/(1-t)^2$ contributes exactly with $i-a+2$ to this coefficient, the term $-(t^a q_T(t)+q_{T,\mathcal{M}}(t))/(1-t)$ with at least $-g-g(\mathcal{M})$ and the term $q_T(t)q_{T,\mathcal{M}}(t)$ with a nonnegative number. All in all we get that the coefficient of t^{i+1} in $p_T(t)p_{T,\mathcal{M}}(t)$ is at least $i-a+2-g-g(\mathcal{M})$. The lemma now follows from the previous lemma. □

Shifted order modules

- Given an order module $\mathcal{M} = (\cup_i M_i, \varphi)$, we can shift the order module by s as follows: $\mathcal{M}_{+s} = (\cup_i M_{i+s}, \varphi)$. Then $\nu(T, \mathcal{M}_{+s}, i) = \nu(T, \mathcal{M}, i+s)$ implying that $\nu(T, \mathcal{M}, s) = \nu(T, \mathcal{M}_{+s}, 0)$. Therefore it will be practical to simplify our notation when $i = 0$ by defining:

$$N(T, \mathcal{M}) := N(T, \mathcal{M}, 0), \quad \nu(T, \mathcal{M}) := \nu(T, \mathcal{M}, 0).$$

Shifted order modules

- Given an order module $\mathcal{M} = (\cup_i M_i, \varphi)$, we can shift the order module by s as follows: $\mathcal{M}_{+s} = (\cup_i M_{i+s}, \varphi)$. Then $\nu(T, \mathcal{M}_{+s}, i) = \nu(T, \mathcal{M}, i+s)$ implying that $\nu(T, \mathcal{M}, s) = \nu(T, \mathcal{M}_{+s}, 0)$. Therefore it will be practical to simplify our notation when $i = 0$ by defining:

$$N(T, \mathcal{M}) := N(T, \mathcal{M}, 0), \quad \nu(T, \mathcal{M}) := \nu(T, \mathcal{M}, 0).$$

- We now have the necessary notation to formulate the following proposition that is essential in order to obtain lower bounds on the minimum distance of codes coming from order modules.

Preparation of the generalized order bound

Proposition

Let $\mathcal{M} = (M, \varphi)$ *be an order module for* $R(T)$ *and let* $\mathbf{c} \in \varphi(M_i)^{\perp} \backslash \varphi(M_{i+1})^{\perp}$. *Then* $\mathrm{wt}\,(\mathbf{c}) \geq \nu(T, \mathcal{M}, i)$, *with* $\mathrm{wt}\,(\mathbf{c})$ *the Hamming weight of* $\mathbf{c}$.

Preparation of the generalized order bound

Proposition

Let $\mathcal{M} = (M, \varphi)$ be an order module for $R(T)$ and let $\mathbf{c} \in \varphi(M_i)^\perp \backslash \varphi(M_{i+1})^\perp$. Then $\mathrm{wt}\,(\mathbf{c}) \geq \nu(T, \mathcal{M}, i)$, with $\mathrm{wt}\,(\mathbf{c})$ the Hamming weight of $\mathbf{c}$.

Proof:

- Let $\mathbf{c} = (c_1, \ldots, c_n) \in \varphi(M_i)^\perp \backslash \varphi(M_{i+1})^\perp$. We denote by $\mathbf{D_c}$ the diagonal matrix with $c_1, \ldots, c_n$ on its diagonal.

Preparation of the generalized order bound

Proposition

Let $\mathcal{M} = (M, \varphi)$ be an order module for $R(T)$ and let $\mathbf{c} \in \varphi(M_i)^\perp \backslash \varphi(M_{i+1})^\perp$. Then $\mathrm{wt}\,(\mathbf{c}) \geq \nu(T, \mathcal{M}, i)$, with $\mathrm{wt}\,(\mathbf{c})$ the Hamming weight of $\mathbf{c}$.

Proof:

- Let $\mathbf{c} = (c_1, \ldots, c_n) \in \varphi(M_i)^\perp \backslash \varphi(M_{i+1})^\perp$. We denote by $\mathbf{D_c}$ the diagonal matrix with $c_1, \ldots, c_n$ on its diagonal.
- Let $H(T) = \{\rho_1, \rho_2, \ldots\}$, such that $\rho_k < \rho_l$ if $k < l$. For every $\rho_k \in H(T)$ we choose a function $f_k \in R(T)$ such that $\rho_T(f_k) = \rho_k$. Further we define $v_k := \mathrm{Ev}_D\,(f_k)$. Let N be a natural number such that $\mathrm{Ev}_D\,(L(NT)) = \mathbb{F}^n$ and $N > \max\{k \,|\, (\rho_k, l) \in N(T, \mathcal{M}, i)\}$.

Preparation of the generalized order bound

- Let $\mathbf{H}_1$ be the $N \times n$ matrix whose k-th row is $\mathrm{Ev}_D(f_k)$ for $1 \leq k \leq N$. By choice of N, we have that $\mathrm{rank}\,\mathbf{H}_1 = n$. By item 2 in Definition 8, there exists an integer N_1 such that $M_{N_1} = 0$.

Preparation of the generalized order bound

- Let $\mathbf{H}_1$ be the $N \times n$ matrix whose k-th row is $\mathrm{Ev}_D\,(f_k)$ for $1 \leq k \leq N$. By choice of N, we have that $\mathrm{rank}\,\mathbf{H}_1 = n$. By item 2 in Definition 8, there exists an integer N_1 such that $M_{N_1} = 0$.
- Since φ is assumed to be a surjective linear map to $\mathbb{F}^n$, there exists an N_2 such that $\varphi(M_{N_2}) = \mathbb{F}^n$ and $N_2 > \max\{l \mid (\rho_k, l) \in N(T, \mathcal{M}, i)\}$.

Preparation of the generalized order bound

- Let $\mathbf{H}_1$ be the $N \times n$ matrix whose k-th row is $\mathrm{Ev}_D(f_k)$ for $1 \leq k \leq N$. By choice of N, we have that $\mathrm{rank}\,\mathbf{H}_1 = n$. By item 2 in Definition 8, there exists an integer N_1 such that $M_{N_1} = 0$.
- Since φ is assumed to be a surjective linear map to $\mathbb{F}^n$, there exists an N_2 such that $\varphi(M_{N_2}) = \mathbb{F}^n$ and $N_2 > \max\{l \,|\, (\rho_k, l) \in N(T, \mathcal{M}, i)\}$.
- The set $H(T, \mathcal{M}) \cap [N_1, N_2]$ consists of finitely many integers, say $s_1, \ldots, s_L$. Then we can choose $m_k \in M_{s_k} \backslash M_{s_k - 1}$.
- By the choice of the m_k we see that $\rho_{T,\mathcal{M}}(m_k) < \rho_{T,\mathcal{M}}(m_l)$ if $k < l$.

Preparation of the generalized order bound

- Let $\mathbf{H}_1$ be the $N \times n$ matrix whose k-th row is $\mathrm{Ev}_D\,(f_k)$ for $1 \leq k \leq N$. By choice of N, we have that $\operatorname{rank} \mathbf{H}_1 = n$. By item 2 in Definition 8, there exists an integer N_1 such that $M_{N_1} = 0$.
- Since φ is assumed to be a surjective linear map to $\mathbb{F}^n$, there exists an N_2 such that $\varphi(M_{N_2}) = \mathbb{F}^n$ and $N_2 > \max\{l \,|\, (\rho_k, l) \in N(T, \mathcal{M}, i)\}$.
- The set $H(T, \mathcal{M}) \cap [N_1, N_2]$ consists of finitely many integers, say $s_1, \ldots, s_L$. Then we can choose $m_k \in M_{s_k} \backslash M_{s_k - 1}$.
- By the choice of the m_k we see that $\rho_{T,\mathcal{M}}(m_k) < \rho_{T,\mathcal{M}}(m_l)$ if $k < l$.
- Now we define $h_k := \varphi(m_k)$ and $\mathbf{H}_2$ the $L \times n$ matrix with h_k as k-th row. By our choice of N_1, N_2 and by item 5 in Definition 8, we have that $\operatorname{rank} \mathbf{H}_2 = n$.

Preparation of the generalized order bound

- Consider the matrix $\mathbf{S}(\mathbf{c}) := \mathbf{H}_1\mathbf{D}_\mathbf{c}\mathbf{H}_2^t$. Since $\mathbf{H}_1$ and $\mathbf{H}_2$ have full rank, we see that $\operatorname{rank}\mathbf{S}(\mathbf{c}) = \operatorname{wt}(\mathbf{c})$. We will also show that $\operatorname{rank}\mathbf{S}(\mathbf{c}) \geq \nu(T, \mathcal{M}, i)$.

Preparation of the generalized order bound

- Consider the matrix $\mathbf{S}(\mathbf{c}) := \mathbf{H}_1\mathbf{D}_\mathbf{c}\mathbf{H}_2^t$. Since $\mathbf{H}_1$ and $\mathbf{H}_2$ have full rank, we see that $\operatorname{rank}\mathbf{S}(\mathbf{c}) = \operatorname{wt}(\mathbf{c})$. We will also show that $\operatorname{rank}\mathbf{S}(\mathbf{c}) \geq \nu(T, \mathcal{M}, i)$.
- We have

$$\mathbf{S}(\mathbf{c})_{ij} = \sum_{\lambda=1}^{n} f_i(P_\lambda)c_\lambda\varphi(m_j)_\lambda = \sum_{\lambda=1}^{n} c_\lambda\varphi(f_i m_j)_\lambda = \langle \mathbf{c}, \varphi(f_i m_j)\rangle. \tag{17}$$

Let $(\rho_i, j) \in N(T, \mathcal{M}, i)$. By our choice of N we have that $i \leq N$ and therefore v_i occurs as a row in H_1. Similarly h_j occurs as a row in H_2.

Preparation of the generalized order bound

- Consider the matrix $\mathbf{S}(\mathbf{c}) := \mathbf{H}_1 \mathbf{D}_\mathbf{c} \mathbf{H}_2^t$. Since $\mathbf{H}_1$ and $\mathbf{H}_2$ have full rank, we see that $\operatorname{rank} \mathbf{S}(\mathbf{c}) = \operatorname{wt}(\mathbf{c})$. We will also show that $\operatorname{rank} \mathbf{S}(\mathbf{c}) \geq \nu(T, \mathcal{M}, i)$.
- We have

$$\mathbf{S}(\mathbf{c})_{ij} = \sum_{\lambda=1}^{n} f_i(P_\lambda) c_\lambda \varphi(m_j)_\lambda = \sum_{\lambda=1}^{n} c_\lambda \varphi(f_i m_j)_\lambda = \langle \mathbf{c}, \varphi(f_i m_j) \rangle. \tag{17}$$

  Let $(\rho_i, j) \in N(T, \mathcal{M}, i)$. By our choice of N we have that $i \leq N$ and therefore v_i occurs as a row in H_1. Similarly h_j occurs as a row in H_2.
- Now let $t := \nu(T, \mathcal{M}, i)$ and suppose that

$$N(T, \mathcal{M}, i) = \{(\rho_{i_1}, j_t), (\rho_{i_2}, j_{t-1}), \ldots, (\rho_{i_t}, j_1)\}.$$

Preparation of the generalized order bound

- For convenience, we define $\sigma_k := \rho_{i_k}$. Without loss of generality we can assume that $i_1 < i_2 < \cdots < i_t$. This implies that $j_1 < j_2 < \cdots < j_t$, since if both $k < l$ and $j_k > j_l$, then

$$i + 1 = \sigma_{t+1-l} + j_l < \sigma_{t+1-k} + j_l < \sigma_{t+1-k} + j_k = i + 1.$$

Preparation of the generalized order bound

- For convenience, we define $\sigma_k := \rho_{i_k}$. Without loss of generality we can assume that $i_1 < i_2 < \cdots < i_t$. This implies that $j_1 < j_2 < \cdots < j_t$, since if both $k < l$ and $j_k > j_l$, then

$$i + 1 = \sigma_{t+1-l} + j_l < \sigma_{t+1-k} + j_l < \sigma_{t+1-k} + j_k = i + 1.$$

- Let $\mathbf{H}$ be the $t \times t$ matrix obtained from $\mathbf{S}(\mathbf{c})$ by choosing all those entries $\mathbf{S}(\mathbf{c})_{ij}$ with $i \in \{i_1, \ldots, i_t\}$ and $j \in \{j_1, \ldots, j_t\}$. Clearly $\operatorname{rank} \mathbf{S}(\mathbf{c}) \geq \operatorname{rank} \mathbf{H}$, so the proposition follows if we show that $\mathbf{H}$ has full rank.

Preparation of the generalized order bound

- For convenience, we define $\sigma_k := \rho_{i_k}$. Without loss of generality we can assume that $i_1 < i_2 < \cdots < i_t$. This implies that $j_1 < j_2 < \cdots < j_t$, since if both $k < l$ and $j_k > j_l$, then

$$i + 1 = \sigma_{t+1-l} + j_l < \sigma_{t+1-k} + j_l < \sigma_{t+1-k} + j_k = i + 1.$$

- Let $\mathbf{H}$ be the $t \times t$ matrix obtained from $\mathbf{S}(\mathbf{c})$ by choosing all those entries $\mathbf{S}(\mathbf{c})_{ij}$ with $i \in \{i_1, \ldots, i_t\}$ and $j \in \{j_1, \ldots, j_t\}$. Clearly $\mathrm{rank}\, \mathbf{S}(\mathbf{c}) \geq \mathrm{rank}\, \mathbf{H}$, so the proposition follows if we show that $\mathbf{H}$ has full rank.
- Suppose that $k + l < t + 1$. Then $\varphi(f_{i_k} m_{j_l}) \in \varphi(M_i)$, since $\rho_{T,\mathcal{M}}(f_{i_k} m_{j_l}) = \rho_T(f_{i_k}) + \rho_{T,\mathcal{M}}(m_{j_l}) = \sigma_k + j_l <$ $\sigma_k + j_{t+1-k} = i + 1$.

Preparation of the generalized order bound

- By equation (17) this implies that

$$\mathbf{S}(\mathbf{c})_{i_k j_l} = \langle \mathbf{c}, \varphi(f_{i_k} m_{j_l}) \rangle = 0.$$

Preparation of the generalized order bound

- By equation (17) this implies that

$$\mathbf{S}(\mathbf{c})_{i_k j_l} = \langle \mathbf{c}, \varphi(f_{i_k} m_{j_l}) \rangle = 0.$$

- If $k + l = t + 1$, then a similar computation shows that $\varphi(f_{i_k} m_{j_l}) \in \varphi(M_{i+1})$ and that $\mathbf{S}(\mathbf{c})_{i_k j_l} \neq 0$. This means that $\mathbf{H}$ is of the form

$$\mathbf{H} = \begin{pmatrix} 0 & & * \\ & \iddots & \\ * & & \end{pmatrix},$$

where a $*$ denotes a nonzero element of $\mathbb{F}$.

Preparation of the generalized order bound

- By equation (17) this implies that

$$\mathbf{S}(\mathbf{c})_{i_k j_l} = \langle \mathbf{c}, \varphi(f_{i_k} m_{j_l}) \rangle = 0.$$

- If $k + l = t + 1$, then a similar computation shows that $\varphi(f_{i_k} m_{j_l}) \in \varphi(M_{i+1})$ and that $\mathbf{S}(\mathbf{c})_{i_k j_l} \neq 0$. This means that $\mathbf{H}$ is of the form

$$\mathbf{H} = \begin{pmatrix} 0 & & * \\ & \iddots & \\ * & & \end{pmatrix},$$

where a $*$ denotes a nonzero element of $\mathbb{F}$.

- Thus $\operatorname{rank} \mathbf{H} = t$.

The generalized order bound

- When using the above proposition, one needs to choose an order module. For example for the code $C_L(D, G)$ we could choose the module $\mathcal{M}_\Omega(D, G, T)$ and for the code $C_\Omega(D, G)$, we can use the module $\mathcal{M}_L(D, G, T)$.

The generalized order bound

- When using the above proposition, one needs to choose an order module. For example for the code $C_L(D,G)$ we could choose the module $\mathcal{M}_\Omega(D,G,T)$ and for the code $C_\Omega(D,G)$, we can use the module $\mathcal{M}_L(D,G,T)$.
- Now we describe the generalized order bound. Let $D = P_1 + \cdots + P_n$ as usual and G a divisor such that $\mathrm{supp}\, G \cap \mathrm{supp}\, D = \varnothing$. Suppose that the set $\{T_1, T_2, \ldots, \}$ consists of rational points that do not occur in $\mathrm{supp}\, D$.

The generalized order bound

- When using the above proposition, one needs to choose an order module. For example for the code $C_L(D, G)$ we could choose the module $\mathcal{M}_\Omega(D, G, T)$ and for the code $C_\Omega(D, G)$, we can use the module $\mathcal{M}_L(D, G, T)$.
- Now we describe the generalized order bound. Let $D = P_1 + \cdots + P_n$ as usual and G a divisor such that $\mathrm{supp}\, G \cap \mathrm{supp}\, D = \varnothing$. Suppose that the set $\{T_1, T_2, \dots,\}$ consists of rational points that do not occur in $\mathrm{supp}\, D$.
- Let $S = (S_1, S_2, \dots)$ be a sequence of points, each of which is contained in $\{T_1, T_2, \dots,\}$.

The generalized order bound

- When using the above proposition, one needs to choose an order module. For example for the code $C_L(D,G)$ we could choose the module $\mathcal{M}_\Omega(D,G,T)$ and for the code $C_\Omega(D,G)$, we can use the module $\mathcal{M}_L(D,G,T)$.
- Now we describe the generalized order bound. Let $D = P_1 + \cdots + P_n$ as usual and G a divisor such that $\operatorname{supp} G \cap \operatorname{supp} D = \varnothing$. Suppose that the set $\{T_1, T_2, \ldots, \}$ consists of rational points that do not occur in $\operatorname{supp} D$.
- Let $S = (S_1, S_2, \ldots)$ be a sequence of points, each of which is contained in $\{T_1, T_2, \ldots, \}$.
- We also recursively define the divisors $G_0 := G$, $G_{i+1} := G_i + S_{i+1}$, $H_0 := G$, $H_{i+1} := H_i - S_{i+1}$ and modules

$$\mathcal{M}_S(i) := \mathcal{M}_\Omega(D, H_i, S_{i+1}), \quad \mathcal{M}_S^\perp(i) := \mathcal{M}_L(D, G_i, S_{i+1}).$$

The generalized order bound

With this notation we introduce

$$d_S(G) := \min_{i:i\geq 0, C_L(D,H_i)\neq C_L(D,H_{i+1})} \{\nu(S_{i+1}, \mathcal{M}_S(i))\},$$

$$d_S^\perp(G) := \min_{i:i\geq 0, C_\Omega(D,G_i)\neq C_\Omega(D,G_{i+1})} \{\nu(S_{i+1}, \mathcal{M}_S^\perp(i))\}.$$

The generalized order bound

With this notation we introduce

$$d_S(G) := \min_{i:i\geq 0, C_L(D,H_i)\neq C_L(D,H_{i+1})} \{\nu(S_{i+1}, \mathcal{M}_S(i))\},$$

$$d_S^{\perp}(G) := \min_{i:i\geq 0, C_\Omega(D,G_i)\neq C_\Omega(D,G_{i+1})} \{\nu(S_{i+1}, \mathcal{M}_S^{\perp}(i))\}.$$

Theorem (Generalized Order Bound)

Let $\{T_1, T_2, \ldots\}$ be a rational points not occurring in $\operatorname{supp} D$ and let $S = (S_1, S_2, \ldots)$ be a subsequence. Then

- *min. dist. of* $C_L(D, G) = d \geq d_S(G)$,
- *min. dist. of* $C_\Omega(D, G) = d^{\perp} \geq d_S^{\perp}(G)$.

Proof of the generalized order bound

Proof:

- We will prove the statements about the code $C_L(D, G)$. The results for the code $C_\Omega(D, G)$ can be proved similarly.

Proof of the generalized order bound

Proof:

- We will prove the statements about the code $C_L(D,G)$. The results for the code $C_\Omega(D,G)$ can be proved similarly.
- Recall that $\nu(T,\mathcal{M}) := \nu(T,\mathcal{M},0)$. We can write $C_L(D,G)$ as the disjoint union $\cup_{i\geq 0} C_L(D,H_i)\backslash C_L(D,H_{i+1})$. If $C_L(D,H_i) \neq C_L(D,H_{i+1})$ and $\mathbf{c} \in C_L(D,H_i)\backslash C_L(D,H_{i+1})$, then from Proposition 3 we see that $\mathrm{wt}\,(\mathbf{c}) \geq \nu(S_{i+1},\mathcal{M}_S(i))$. Then it follows that $d \geq \min_i\{\nu(S_{i+1},\mathcal{M}_S(i))\}$, if we take the minimum over all nonnegative i such that $C_L(D,H_i) \neq C_L(D,H_{i+1})$.

$\square$

The Goppa–bound

As a special case of the generalized order bound we get:

Corollary (The Goppa–bound)

- $\textit{min. dist. of } C_L(D, G) = d \geq n - \deg G,$
- $\textit{min. dist. of } C_\Omega(D, G) = d^\perp \geq \deg G - 2g + 2.$

The Goppa–bound

As a special case of the generalized order bound we get:

Corollary (The Goppa–bound)

- *min. dist. of* $C_L(D,G) = d \geq n - \deg G$,
- *min. dist. of* $C_\Omega(D,G) = d^\perp \geq \deg G - 2g + 2$.

Proof:

- $\mathcal{M}_S(i) = \mathcal{M}_\Omega(D, H_i, S_{i+1})$ and $H_i = G - S_0 - \cdots - S_i$. Using the notion of gaps and the above lemma gives

$$\nu(S_{i+1}, \mathcal{M}_S(i)\,) \geq n - \deg G + i \geq n - \deg G.$$

Therefore $d \geq d_S(G) \geq n - \deg G$.

The Goppa–bound

- Similarly it holds that

$$\nu(S_{i+1}, \mathcal{M}_S^{\perp}(i)) \geq \deg G + i - 2g + 2 \geq \deg G - 2g + 2,$$

which implies that $d^{\perp} \geq d_S^{\perp}(G) \geq \deg G - 2g + 2$.

$\square$

The Goppa–bound

- Similarly it holds that

$$\nu(S_{i+1}, \mathcal{M}_S^{\perp}(i)) \geq \deg G + i - 2g + 2 \geq \deg G - 2g + 2,$$

which implies that $d^{\perp} \geq d_S^{\perp}(G) \geq \deg G - 2g + 2$.

☐It is time for an example again!

The Goppa–bound

- Similarly it holds that

$$\nu(S_{i+1}, \mathcal{M}_S^{\perp}(i)) \geq \deg G + i - 2g + 2 \geq \deg G - 2g + 2,$$

which implies that $d^{\perp} \geq d_S^{\perp}(G) \geq \deg G - 2g + 2$.

□It is time for an example again!

Example In this example we will study a code coming from the Hermitian curve defined over $\mathbb{F}_{64}$ by the equation

$$x_2^8 + x_2 = x_1^9.$$

This curve has 513 rational points, exactly one of which is a common pole of x_1 and x_2.

Example

- As usual, we denote this point by T_∞. We denote by T_0 the unique point having a zero in both x_1 and x_2. Further, we denote by D the sum of the 504 rational points P satisfying $x_1(P) \neq 0$.

Example

- As usual, we denote this point by T_∞. We denote by T_0 the unique point having a zero in both x_1 and x_2. Further, we denote by D the sum of the 504 rational points P satisfying $x_1(P) \neq 0$.
- In this example we will consider the code $C_L(D, -T_0 + 490T_\infty)$. This is a $[504, 462, \geq 15]$ code, since $l(-T_0 + 490T_\infty) = 462$ and the Goppa bound gives that the minimum distance is at least $504 - 489 = 15$. We will show that the Goppa bound is not sharp in this case and show that the minimum distance is at least 21.

Example

- As usual, we denote this point by T_∞. We denote by T_0 the unique point having a zero in both x_1 and x_2. Further, we denote by D the sum of the 504 rational points P satisfying $x_1(P) \neq 0$.
- In this example we will consider the code $C_L(D, -T_0 + 490T_\infty)$. This is a $[504, 462, \geq 15]$ code, since $l(-T_0 + 490T_\infty) = 462$ and the Goppa bound gives that the minimum distance is at least $504 - 489 = 15$. We will show that the Goppa bound is not sharp in this case and show that the minimum distance is at least 21.
- We wish to use Theorem 12 to get a lower bound on the minimum distance of the code $C_L(D, -T_0 + 490T_\infty)$.

Example

- First we need to choose a sequence S, which we take to be $S := (T_\infty, T_0, T_0, T_0, \dots)$ in this example. We will compute the quantity $d_S(-T_0 + 490T_\infty)$. In order to do so we will work in the modules $\mathcal{M}^{(i)}{}_\Omega(S)$.

Example

- First we need to choose a sequence S, which we take to be $S := (T_\infty, T_0, T_0, T_0, \dots)$ in this example. We will compute the quantity $d_S(-T_0 + 490T_\infty)$. In order to do so we will work in the modules $\mathcal{M}^{(i)}{}_\Omega(S)$.
- The first module we need to work in is $\mathcal{M}_S(0) = \mathcal{M}_\Omega(D, -T_0 + 490T_\infty, T_\infty)$. We start by calculating $H(T_\infty, \mathcal{M}_S(0))$.

Example

- First we need to choose a sequence S, which we take to be $S := (T_\infty, T_0, T_0, T_0, \ldots)$ in this example. We will compute the quantity $d_S(-T_0 + 490T_\infty)$. In order to do so we will work in the modules $\mathcal{M}^{(i)}{}_\Omega(S)$.
- The first module we need to work in is $\mathcal{M}_S(0) = \mathcal{M}_\Omega(D, -T_0 + 490T_\infty, T_\infty)$. We start by calculating $H(T_\infty, \mathcal{M}_S(0))$.
- We will need to know what $\rho_{T_\infty}(\Omega(-D - T_0 + 490T_\infty))$ is. The Weierstrass semigroup $H(T_\infty)$ is generated by 8 and 9, i.e. $H(T_\infty) = \langle 8, 9\rangle = \{0, 8, 9, 16, 17, 18, 24, \ldots\}$.

Example

- First we need to choose a sequence S, which we take to be $S := (T_\infty, T_0, T_0, T_0, \dots)$ in this example. We will compute the quantity $d_S(-T_0 + 490T_\infty)$. In order to do so we will work in the modules $\mathcal{M}^{(i)}{}_\Omega(S)$.
- The first module we need to work in is $\mathcal{M}_S(0) = \mathcal{M}_\Omega(D, -T_0 + 490T_\infty, T_\infty)$. We start by calculating $H(T_\infty, \mathcal{M}_S(0))$.
- We will need to know what $\rho_{T_\infty}(\Omega(-D - T_0 + 490T_\infty))$ is. The Weierstrass semigroup $H(T_\infty)$ is generated by 8 and 9, i.e. $H(T_\infty) = \langle 8, 9\rangle = \{0, 8, 9, 16, 17, 18, 24, \dots\}$.
- It holds that $H(T) = H(T_\infty)$ for any rational point T. This means that the Laurent series $p(t) := \sum_{i \in \langle 8,9\rangle} t^i$ will play a central role in the following.

Example

- For any order module and for any $m \in M_i \backslash M_{i-1}$ we have $\rho_{T,\mathcal{M}}(m) = i$. We see that for $m \in \Omega(-D - T_0 + (490 - i)T_\infty) \backslash \Omega(-D - T_0 + (491 - i)T_\infty)$ we have $\rho_{T_\infty,\mathcal{M}_S(0)}(m) = \rho_{T_\infty}(m) + 490$. Further, using the differential $\omega = (x_1^{63} + 1)^{-1} dx_1$, we see that

$$\rho_{T_\infty}(\Omega(-D - T_0 + (490 - i)T_\infty)) = \{-558 + s \mid s \in \rho_{T_\infty}(L(T_0 + (68 + i)$$

Example

- For any order module and for any $m \in M_i \backslash M_{i-1}$ we have $\rho_{T,\mathcal{M}}(m) = i$. We see that for $m \in \Omega(-D - T_0 + (490 - i)T_\infty) \backslash \Omega(-D - T_0 + (491 - i)T_\infty)$ we have $\rho_{T_\infty,\mathcal{M}_S(0)}(m) = \rho_{T_\infty}(m) + 490$. Further, using the differential $\omega = (x_1^{63} + 1)^{-1} dx_1$, we see that

$$\rho_{T_\infty}(\Omega(-D-T_0+(490-i)T_\infty)) = \{-558+s \mid s \in \rho_{T_\infty}(L(T_0+(68+i)$$

- Using the description of L-spaces in Example 1 from before, we see that

$$\bigcup_{i \in \mathbb{Z}} \rho_{T_\infty}(L(T_0 + (68 + i)T_\infty)) = H(T_\infty) \cup \{55\}.$$

Putting everything together, we find that

$$H(T_\infty, \mathcal{M}_S(0)) = \{s - 68 \mid s \in H(T_\infty)\} \cup \{-13\}.$$

Example

- Therefore
$$p_{T_\infty,\mathcal{M}_S(0)}(t) = t^{-13} + t^{-68}p(t)$$
Using equation the expansion of $p(t)$, we get
$$p(t)p_{T_\infty,\mathcal{M}_S(0)}(t) = \cdots + 24t + 21t^2 + 17t^3 + \cdots ,$$
and therefore (see Lemma 10): $\nu(T_\infty, \mathcal{M}_S(0)) = 24$.

Example

- Therefore

$$p_{T_\infty,\mathcal{M}_S(0)}(t) = t^{-13} + t^{-68}p(t)$$

Using equation the expansion of $p(t)$, we get

$$p(t)p_{T_\infty,\mathcal{M}_S(0)}(t) = \cdots + 24t + 21t^2 + 17t^3 + \cdots ,$$

and therefore (see Lemma 10): $\nu(T_\infty, \mathcal{M}_S(0)) = 24$.

- For the next step we need to know the set $H(T_0, \mathcal{M}_S(1))$. Note that $H(T_0) = H(T_\infty)$. We will calculate $\rho_{T_0}(L((1+i)T_0 + 69T_\infty))$.
- Using the fact that $(x_2) = 9(T_0 - T_\infty)$, we see that

$$\rho_{T_0}(L((1+i)T_0+69T_\infty)) = \{s-63 | s \in \rho_{T_0}(L((64+i)T_0+6T_\infty))\}.$$

Example

- The automorphism τ defined by $\tau(x_1) = x_1/x_2$ and $\tau(x_2) = 1/x_2$, interchanges the points T_0 and T_∞. Using this automorphism, we can conclude that

$$\rho_{T_0}(L((64+i)T_0 + 6T_\infty)) = \rho_{T_\infty}(L((64+i)T_\infty + 6T_0)).$$

Example

- The automorphism τ defined by $\tau(x_1) = x_1/x_2$ and $\tau(x_2) = 1/x_2$, interchanges the points T_0 and T_∞. Using this automorphism, we can conclude that

$$\rho_{T_0}(L((64+i)T_0 + 6T_\infty)) = \rho_{T_\infty}(L((64+i)T_\infty + 6T_0)).$$

- Similarly we find that $H(T_0, \mathcal{M}_S(1))$ equals

$$\{s-64 \mid s \in H(T_0)\} \cup \{-49, -41, -33, -25, -17, -9\}.$$

This implies that

$$p_{T_0,\mathcal{M}_S(1)}(t) = t^{-49} + t^{-41} + t^{-33} + t^{-25} + t^{-17} + t^{-9} + t^{-64}p(t),$$

enabling us to calculate that

$$p(t)p_{T_0,\mathcal{M}_S(1)}(t) = \cdots + 21t + 25t^2 + 27t^3 + 27t^4 + 25t^5 + \cdots .$$

Example

- Hence $\nu(T_0, \mathcal{M}_S(1)) = 21$. Since the sequence S only contains T_0 apart from the very first point in the sequence, it suffices to work with the module $\mathcal{M}_S(1)$.

Example

- Hence $\nu(T_0, \mathcal{M}_S(1)) = 21$. Since the sequence S only contains T_0 apart from the very first point in the sequence, it suffices to work with the module $\mathcal{M}_S(1)$.
- For $i \geq 0$, we can see the module $\mathcal{M}_S(i+1)$ as the i-th shift of $\mathcal{M}_S(1)$. More precisely, we have that $\nu(T_0, \mathcal{M}_S(i+1)) = \nu(T_0, \mathcal{M}_S(1), i)$. This means that with the above computation of $H(T_0, \mathcal{M}_S(1))$, we have all information we need to calculate $d_S(-T_0 + 490T_\infty)$.

Example

- Hence $\nu(T_0, \mathcal{M}_S(1)) = 21$. Since the sequence S only contains T_0 apart from the very first point in the sequence, it suffices to work with the module $\mathcal{M}_S(1)$.
- For $i \geq 0$, we can see the module $\mathcal{M}_S(i+1)$ as the i-th shift of $\mathcal{M}_S(1)$. More precisely, we have that $\nu(T_0, \mathcal{M}_S(i+1)) = \nu(T_0, \mathcal{M}_S(1), i)$. This means that with the above computation of $H(T_0, \mathcal{M}_S(1))$, we have all information we need to calculate $d_S(-T_0 + 490T_\infty)$.
- We see from the equation on the previous slide that $\nu(T_0, \mathcal{M}_S(2)) = \nu(T_0, \mathcal{M}_S(5)) = 25$ and $\nu(T_0, \mathcal{M}_S(3)) = \nu(T_0, \mathcal{M}_S(4)) = 27$. For $i \geq 6$, we can use Lemma 11 to show that $\nu(T_0, \mathcal{M}_S(i)) \geq 15 + i \geq 21$.
- All in all, we have shown that $d_S(-T_0 + 490T_\infty) = 21$.

Contents

Majority voting

- For a code $C_L(D, G)$, the basic algorithm can correct $\lfloor (n - \deg G - 1 - g)/2 \rfloor$ errors. This means that the full potential of the code has not been used yet.
- We will describe an algorithm that can correct $\lfloor (d_S(G) - 1)/2 \rfloor$ errors, where $d_S(G)$ denotes the generalized order bound.
- This is achieved using *majority voting* for so-called unknown syndromes.
- Loosely speaking this technique enables one to obtain more information about the error-vector, and thereby to correct more errors than with the basic algorithm.

Syndromes and syndrome matrix

- Let $\mathbf{r} = \mathbf{c} + \mathbf{e}$. The fact that for the $(n - l_0) \times l_1$ matrix $\mathbf{S}^{(A)}(\mathbf{r})$ we have that $\mathbf{S}^{(A)}(\mathbf{c}) = \mathbf{S}^{(A)}(\mathbf{e})$ is central in showing that the basic algorithm can correct $\lfloor (n - \deg G - 1 - g)/2 \rfloor$ errors.

Syndromes and syndrome matrix

- Let $\mathbf{r} = \mathbf{c} + \mathbf{e}$. The fact that for the $(n - l_0) \times l_1$ matrix $\mathbf{S}^{(A)}(\mathbf{r})$ we have that $\mathbf{S}^{(A)}(\mathbf{c}) = \mathbf{S}^{(A)}(\mathbf{e})$ is central in showing that the basic algorithm can correct $\lfloor (n - \deg G - 1 - g)/2 \rfloor$ errors.
- The matrix $\mathbf{S}^{(A)}(\mathbf{r})$ therefore gives information about the error-vector $\mathbf{e}$. In fact, we know that its kernel determines the error-locator Q_1.

Syndromes and syndrome matrix

- Let $\mathbf{r} = \mathbf{c} + \mathbf{e}$. The fact that for the $(n - l_0) \times l_1$ matrix $\mathbf{S}^{(A)}(\mathbf{r})$ we have that $\mathbf{S}^{(A)}(\mathbf{c}) = \mathbf{S}^{(A)}(\mathbf{e})$ is central in showing that the basic algorithm can correct $\lfloor (n - \deg G - 1 - g)/2 \rfloor$ errors.
- The matrix $\mathbf{S}^{(A)}(\mathbf{r})$ therefore gives information about the error-vector $\mathbf{e}$. In fact, we know that its kernel determines the error-locator Q_1.

Definition (Unknown syndrome)

If ω and h are such that $h\omega \notin \Omega(-D + G)$, then the syndrome $s_{\omega,h}(\mathbf{r})$ will in general depend both on $\mathbf{c}$ and $\mathbf{e}$. Such a syndrome it said to be *unknown*.

Syndromes and syndrome matrix

Definition (Syndrome)

Let ω be a differential form. Then we define

$$s_\omega(\mathbf{r}) := s_{\omega,1}(\mathbf{r}).$$

- Let $T \notin \operatorname{supp} G$ be a rational point. For now let us assume that $A = G + aT$.
- We can do this, since the only restrictions on A were that $\deg A < n - t$ and $l(A - G) > t$. If $t + g - 1 < a < n - t - \deg G$ both conditions are guaranteed to hold.

Syndromes and syndrome matrix

Definition (Syndrome)

Let ω be a differential form. Then we define

$$s_\omega(\mathbf{r}) := s_{\omega,1}(\mathbf{r}).$$

- Let $T \notin \operatorname{supp} G$ be a rational point. For now let us assume that $A = G + aT$.
- We can do this, since the only restrictions on A were that $\deg A < n - t$ and $l(A - G) > t$. If $t + g - 1 < a < n - t - \deg G$ both conditions are guaranteed to hold.
- It will be convenient to extend the matrix $\mathbf{S}^{(A)}(\mathbf{r})$ in this setup.

Syndromes and syndrome matrix

- The matrix $\mathbf{S}^{(A)}(\mathbf{r})$ itself depends on the choice of functions and differentials from $L(A-G)$ and $\Omega(A-D)$.
- We now specify a more precise choice: let $H(T) = \{\rho_1, \rho_2, \dots\}$ and $h_1, h_2, \dots \in R(T)$ such that $\rho_T(h_i) = \rho_i$.
- Similarly, let $\mathcal{M} := \mathcal{M}_\Omega(D, G, T)$ and $H(T, \mathcal{M}) = \{\sigma_1, \sigma_2, \dots\}$.

Syndromes and syndrome matrix

- The matrix $\mathbf{S}^{(A)}(\mathbf{r})$ itself depends on the choice of functions and differentials from $L(A-G)$ and $\Omega(A-D)$.
- We now specify a more precise choice: let $H(T) = \{\rho_1, \rho_2, \dots\}$ and $h_1, h_2, \dots \in R(T)$ such that $\rho_T(h_i) = \rho_i$.
- Similarly, let $\mathcal{M} := \mathcal{M}_\Omega(D, G, T)$ and $H(T, \mathcal{M}) = \{\sigma_1, \sigma_2, \dots\}$.
- We can then choose differential forms $\omega_1, \omega_2, \dots \in \cup_i \Omega(-D + G - iT)$ such that $\rho_{T,\mathcal{M}}(\omega_j) = \sigma_j$. We then define the following matrices: ...

Syndrome matrix

Definition

Let

$$\mathbf{S}_T^{tot}(\mathbf{r}) := \begin{pmatrix} s_{\omega_1,h_1}(\mathbf{r}) & s_{\omega_1,h_2}(\mathbf{r}) & \cdots \\ s_{\omega_2,h_1}(\mathbf{r}) & s_{\omega_2,h_2}(\mathbf{r}) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

and

$$\mathbf{S}_T^{tot}(\mathbf{r})|_{i,j} := \begin{pmatrix} s_{\omega_1,h_1}(\mathbf{r}) & \cdots & s_{\omega_1,h_i}(\mathbf{r}) \\ \vdots & & \vdots \\ s_{\omega_j,h_1}(\mathbf{r}) & \cdots & s_{\omega_j,h_i}(\mathbf{r}) \end{pmatrix}.$$

The matrix $\mathbf{S}_T^{tot}(\mathbf{r})$ extends the matrix $\mathbf{S}^{(A)}(\mathbf{r})$ in the case that $A = G + aT$.

Candidates and discrepancy

- Note that $h_i \omega_j \in \Omega(-D + G - (\rho_i + \sigma_j)T)$. Therefore we have that all elements $s_{\omega_j,h_i}(\mathbf{r})$ of $\mathbf{S}_T^{tot}(\mathbf{r})$ such that $\rho_i + \sigma_j \leq 0$, are known syndromes, i.e. equal to $s_{\omega_j,h_i}(\mathbf{e})$.

Candidates and discrepancy

- Note that $h_i\omega_j \in \Omega(-D + G - (\rho_i + \sigma_j)T)$. Therefore we have that all elements $s_{\omega_j,h_i}(\mathbf{r})$ of $\mathbf{S}_T^{tot}(\mathbf{r})$ such that $\rho_i + \sigma_j \leq 0$, are known syndromes, i.e. equal to $s_{\omega_j,h_i}(\mathbf{e})$.
- Before proceedinging, we need some terminology:

Definition (Candidate and discrepancy)

A position (i,j) in the matrix $\mathbf{S}_T^{tot}(\mathbf{e})$ is said to be a *candidate*, if the matrices $\mathbf{S}_T^{tot}(\mathbf{e})|_{i-1,j-1}$, $\mathbf{S}_T^{tot}(\mathbf{e})|_{i-1,j}$, and $\mathbf{S}_T^{tot}(\mathbf{e})|_{i,j-1}$ all have the same rank.

Candidates and discrepancy

- Note that $h_i\omega_j \in \Omega(-D + G - (\rho_i + \sigma_j)T)$. Therefore we have that all elements $s_{\omega_j,h_i}(\mathbf{r})$ of $\mathbf{S}_T^{tot}(\mathbf{r})$ such that $\rho_i + \sigma_j \leq 0$, are known syndromes, i.e. equal to $s_{\omega_j,h_i}(\mathbf{e})$.
- Before proceedinging, we need some terminology:

Definition (Candidate and discrepancy)

A position (i,j) in the matrix $\mathbf{S}_T^{tot}(\mathbf{e})$ is said to be a *candidate*, if the matrices $\mathbf{S}_T^{tot}(\mathbf{e})|_{i-1,j-1}$, $\mathbf{S}_T^{tot}(\mathbf{e})|_{i-1,j}$, and $\mathbf{S}_T^{tot}(\mathbf{e})|_{i,j-1}$ all have the same rank.
If furthermore the matrices $\mathbf{S}_T^{tot}(\mathbf{e})|_{i-1,j-1}$ and $\mathbf{S}_T^{tot}(\mathbf{e})|_{i,j}$ do not have equal rank, then the position (i,j) is called a *discrepancy*.

Candidates and known syndromes

- Now suppose that $\mathbf{r} = \mathbf{c} + \mathbf{e}$, with $\mathbf{c} \in C_L(D, G)$ and that we are given a candidate (i, j) with $\rho_i + \sigma_j = 1$.
- We can determine these candidates, since the part of the matrix $\mathbf{S}_T^{tot}(\mathbf{e})$ that we need to determine them only involves *known* syndromes.

Candidates and known syndromes

- Now suppose that $\mathbf{r} = \mathbf{c} + \mathbf{e}$, with $\mathbf{c} \in C_L(D, G)$ and that we are given a candidate (i, j) with $\rho_i + \sigma_j = 1$.
- We can determine these candidates, since the part of the matrix $\mathbf{S}_T^{tot}(\mathbf{e})$ that we need to determine them only involves *known* syndromes.
- Furthermore, suppose that $\omega_l \in \Omega(-D + G - T) \backslash \Omega(-D + G)$. Then there exists constants $\mu \in \mathbb{F} \backslash \{0\}$ and $\mu_k \in \mathbb{F}$ (only depending on (i, j)) such that

$$\omega_l = \mu h_i \omega_j + \sum_{k=0}^{l-1} \mu_k \omega_k. \tag{18}$$

Votes

- There exists a unique element $\alpha \in \mathbb{F}$ such that the matrix $\mathbf{M}$ obtained from $\mathbf{S}_T^{tot}(\mathbf{r})|_{i,j}$ by replacing its $(i,j)-th$ element by α, has the same rank as the matrix $\mathbf{S}_T^{tot}(\mathbf{r})|_{i-1,j-1}$.

Votes

- There exists a unique element $\alpha \in \mathbb{F}$ such that the matrix $\mathbf{M}$ obtained from $\mathbf{S}_T^{tot}(\mathbf{r})|_{i,j}$ by replacing its $(i,j)-th$ element by α, has the same rank as the matrix $\mathbf{S}_T^{tot}(\mathbf{r})|_{i-1,j-1}$.
- We say that the candidate (i,j) *votes* for α concerning the syndrome $s_{\omega_j,h_i}(\mathbf{e})$. Using equation (18) we then also get a value for $s_{\omega_l}(\mathbf{e})$.

Votes

- There exists a unique element $\alpha \in \mathbb{F}$ such that the matrix $\mathbf{M}$ obtained from $\mathbf{S}_T^{tot}(\mathbf{r})|_{i,j}$ by replacing its $(i,j)-th$ element by α, has the same rank as the matrix $\mathbf{S}_T^{tot}(\mathbf{r})|_{i-1,j-1}$.
- We say that the candidate (i,j) *votes* for α concerning the syndrome $s_{\omega_j,h_i}(\mathbf{e})$. Using equation (18) we then also get a value for $s_{\omega_l}(\mathbf{e})$.
- If this value is correct, we say that the candidate votes correctly, otherwise we say that the candidate votes incorrectly.

Votes

- There exists a unique element $\alpha \in \mathbb{F}$ such that the matrix $\mathbf{M}$ obtained from $\mathbf{S}_T^{tot}(\mathbf{r})|_{i,j}$ by replacing its $(i,j)-th$ element by α, has the same rank as the matrix $\mathbf{S}_T^{tot}(\mathbf{r})|_{i-1,j-1}$.
- We say that the candidate (i,j) *votes* for α concerning the syndrome $s_{\omega_j,h_i}(\mathbf{e})$. Using equation (18) we then also get a value for $s_{\omega_l}(\mathbf{e})$.
- If this value is correct, we say that the candidate votes correctly, otherwise we say that the candidate votes incorrectly.
- We now show that this voting procedure gives the right value for $s_{\omega_j,h_i}(\mathbf{e})$ in the majority of cases, if we assume that not too many errors have occurred.

Votes

Theorem

- *Let* $\mathbf{r} = \mathbf{c} + \mathbf{e}$ *with* $\mathbf{c} \in C_L(D, G)$.
- *Let* $\omega_l \in \Omega(-D + G - T) \backslash \Omega(-D + G)$ *and assume that* $C_L(D, G) \neq C_L(D, G - T)$ *and that* $2\mathrm{wt}(\mathbf{e}) < \nu(T, \mathcal{M}_\Omega(D, G, T))$.
- *Then the majority of candidates in* $N(T, \mathcal{M}_\Omega(D, G, T))$ *vote for the correct value of* $s_{\omega_l}(\mathbf{e})$.

Votes

Theorem

- *Let* $\mathbf{r} = \mathbf{c} + \mathbf{e}$ *with* $\mathbf{c} \in C_L(D, G)$.
- *Let* $\omega_l \in \Omega(-D + G - T) \setminus \Omega(-D + G)$ *and assume that* $C_L(D, G) \neq C_L(D, G - T)$ *and that* $2\mathrm{wt}(\mathbf{e}) < \nu(T, \mathcal{M}_\Omega(D, G, T))$.
- *Then the majority of candidates in* $N(T, \mathcal{M}_\Omega(D, G, T))$ *vote for the correct value of* $s_{\omega_l}(\mathbf{e})$.

Proof:

We consider the following sets:

Votes

$$\begin{aligned}
\mathtt{K} &:= \{(i,j) \,|\, (i,j) \text{ a discrepancy}, \rho_i + \sigma_j < 1\}, \\
\mathtt{F} &:= \{(i,j) \in N(T,\mathcal{M},0) \,|\, (i,j) \text{ cand. voting incorrectly for } s_{\omega_l}(\mathbf{e})\} \\
\mathtt{T} &:= \{(i,j) \in N(T,\mathcal{M},0) \,|\, (i,j) \text{ cand. voting correctly for } s_{\omega_l}(\mathbf{e})\}.
\end{aligned}$$

Votes

$$\begin{aligned} \mathtt{K} &:= \{(i,j) \mid (i,j) \text{ a discrepancy}, \rho_i + \sigma_j < 1\}, \\ \mathtt{F} &:= \{(i,j) \in N(T,\mathcal{M},0) \mid (i,j) \text{ cand. voting incorrectly for } s_{\omega_l}(\mathbf{e})\} \\ \mathtt{T} &:= \{(i,j) \in N(T,\mathcal{M},0) \mid (i,j) \text{ cand. voting correctly for } s_{\omega_l}(\mathbf{e})\}. \end{aligned}$$

- Let ρ_{N_1} (resp. σ_{N_2}) be the largest first (resp. second) coordinate occurring in $N(T,\mathcal{M},0)$.

Votes

$$\begin{aligned}
\mathtt{K} &:= \{(i,j) \mid (i,j) \text{ a discrepancy}, \rho_i + \sigma_j < 1\}, \\
\mathtt{F} &:= \{(i,j) \in N(T,\mathcal{M},0) \mid (i,j) \text{ cand. voting incorrectly for } s_{\omega_l}(\mathbf{e})\} \\
\mathtt{T} &:= \{(i,j) \in N(T,\mathcal{M},0) \mid (i,j) \text{ cand. voting correctly for } s_{\omega_l}(\mathbf{e})\}.
\end{aligned}$$

- Let ρ_{N_1} (resp. σ_{N_2}) be the largest first (resp. second) coordinate occurring in $N(T,\mathcal{M},0)$.
- The matrix $\mathbf{S}_T^{tot}(\mathbf{e})|_{N_1,N_2}$ has rank $\mathrm{wt}\,(\mathbf{e})$, but on the other hand it is at least $\#\mathtt{K} + \#\mathtt{F}$, since discrepancies are exactly pivot positions in the matrix $\mathbf{S}_T^{tot}(\mathbf{e})|_{N_1,N_2}$.

Votes

$$\begin{aligned} \mathtt{K} &:= \{(i,j) \,|\, (i,j) \text{ a discrepancy}, \rho_i + \sigma_j < 1\}, \\ \mathtt{F} &:= \{(i,j) \in N(T,\mathcal{M},0) \,|\, (i,j) \text{ cand. voting incorrectly for } s_{\omega_l}(\mathbf{e})\} \\ \mathtt{T} &:= \{(i,j) \in N(T,\mathcal{M},0) \,|\, (i,j) \text{ cand. voting correctly for } s_{\omega_l}(\mathbf{e})\}. \end{aligned}$$

- Let ρ_{N_1} (resp. σ_{N_2}) be the largest first (resp. second) coordinate occurring in $N(T,\mathcal{M},0)$.
- The matrix $\mathbf{S}_T^{tot}(\mathbf{e})|_{N_1,N_2}$ has rank $\mathrm{wt}(\mathbf{e})$, but on the other hand it is at least $\#\mathtt{K} + \#\mathtt{F}$, since discrepancies are exactly pivot positions in the matrix $\mathbf{S}_T^{tot}(\mathbf{e})|_{N_1,N_2}$.
- Therefore we have that

$$2\#\mathtt{K} + 2\#\mathtt{F} \leq 2\mathrm{wt}(\mathbf{e}) < \nu(T,\mathcal{M}).$$

Votes

- If an element $(i, j) \in N(T, \mathcal{M}, 0)$ is not a candidate, then there exists an element of $\mathtt{K}$ with first coordinate i or second coordinate j.
- Therefore, the number of non-candidates in $N(T, \mathcal{M}, 0)$ is at most $2\#\mathtt{K}$.
- The number of candidates in $N(T, \mathcal{M}, 0)$ is equal to $\#\mathtt{F} + \#\mathtt{T}$.
- All in all we find that $\nu(T, \mathcal{M}) \leq 2\#\mathtt{K} + \#\mathtt{F} + \#\mathtt{T}$.
- Combining this with the above, we see that $\#\mathtt{T} > \#\mathtt{F}$.

Decoding up to half the generalized order bound

- If $C_L(D, G) = C_L(D, G - T)$, but $\Omega(-D + G - T) \neq \Omega(-D + G)$ then $s_{\omega_l}(\mathbf{e})$ for $\omega_l \in \Omega(-D + G - T)$ can be determined as follows:

Decoding up to half the generalized order bound

- If $C_L(D, G) = C_L(D, G - T)$, but $\Omega(-D + G - T) \neq \Omega(-D + G)$ then $s_{\omega_l}(\mathbf{e})$ for $\omega_l \in \Omega(-D + G - T)$ can be determined as follows:
- There exists $\omega \in \Omega(-D + G)$ such that $\mathrm{Res}_D(\omega) = \mathrm{Res}_D(\omega_l)$, and therefore $s_{\omega_l}(\mathbf{e}) = s_{\omega}(\mathbf{e})$. But the latter is a known syndrome.

Decoding up to half the generalized order bound

- If $C_L(D, G) = C_L(D, G - T)$, but $\Omega(-D + G - T) \neq \Omega(-D + G)$ then $s_{\omega_l}(\mathbf{e})$ for $\omega_l \in \Omega(-D + G - T)$ can be determined as follows:
- There exists $\omega \in \Omega(-D + G)$ such that $\mathrm{Res}_D(\omega) = \mathrm{Res}_D(\omega_l)$, and therefore $s_{\omega_l}(\mathbf{e}) = s_\omega(\mathbf{e})$. But the latter is a known syndrome.
- Combined with the above theorem, we see that we can always determine the value of $s_{\omega_l}(\mathbf{e})$ as long as $2\mathrm{wt}(\mathbf{e}) < \nu(T, \mathcal{M})$.
- The minimum distance d of $C_L(D, G)$ satisfies $d \geq d_S(G) := \min_i\{\nu(S_{i+1}, \mathcal{M}_S(i))\}$, where the minimum is taken over all i such that $C_L(D, H_i) \neq C_L(D, H_{i+1})$.

Decoding up to half the generalized order bound

- If $C_L(D,G) = C_L(D,G-T)$, but $\Omega(-D+G-T) \neq \Omega(-D+G)$ then $s_{\omega_l}(\mathbf{e})$ for $\omega_l \in \Omega(-D+G-T)$ can be determined as follows:
- There exists $\omega \in \Omega(-D+G)$ such that $\mathrm{Res}_D(\omega) = \mathrm{Res}_D(\omega_l)$, and therefore $s_{\omega_l}(\mathbf{e}) = s_\omega(\mathbf{e})$. But the latter is a known syndrome.
- Combined with the above theorem, we see that we can always determine the value of $s_{\omega_l}(\mathbf{e})$ as long as $2\mathrm{wt}(\mathbf{e}) < \nu(T,\mathcal{M})$.
- The minimum distance d of $C_L(D,G)$ satisfies $d \geq d_S(G) := \min_i\{\nu(S_{i+1}, \mathcal{M}_S(i))\}$, where the minimum is taken over all i such that $C_L(D,H_i) \neq C_L(D,H_{i+1})$.
- We can decode the code $C_L(D,G)$ up to half this bound.

Decoding up to half the generalized order bound

- (As before) let $\{T_1, T_2, \ldots, \}$ be rational points that do not occur in $\operatorname{supp} D$, and let $S = (S_1, S_2, \ldots)$ be a subsequence.

Decoding up to half the generalized order bound

- (As before) let $\{T_1, T_2, \ldots, \}$ be rational points that do not occur in $\mathrm{supp}\, D$, and let $S = (S_1, S_2, \ldots)$ be a subsequence.
- Further define divisors $H_0 := G$, $H_{i+1} := H_i - S_{i+1}$ and modules $\mathcal{M}_S(i) := \mathcal{M}_\Omega(D, H_i, S_{i+1})$.

Decoding up to half the generalized order bound

- (As before) let $\{T_1, T_2, \ldots,\}$ be rational points that do not occur in $\operatorname{supp} D$, and let $S = (S_1, S_2, \ldots)$ be a subsequence.
- Further define divisors $H_0 := G$, $H_{i+1} := H_i - S_{i+1}$ and modules $\mathcal{M}_S(i) := \mathcal{M}_\Omega(D, H_i, S_{i+1})$.
- We can determine all unknown syndromes using the previous theorem (majority voting) iteratively on the sequence of codes $C_L(D, G) \supset \cdots \supset C_L(D, H_i) \supset C_L(D, H_{i+1}) \supset \cdots$.

Decoding up to half the generalized order bound

- (As before) let $\{T_1, T_2, \ldots,\}$ be rational points that do not occur in $\operatorname{supp} D$, and let $S = (S_1, S_2, \ldots)$ be a subsequence.
- Further define divisors $H_0 := G$, $H_{i+1} := H_i - S_{i+1}$ and modules $\mathcal{M}_S(i) := \mathcal{M}_\Omega(D, H_i, S_{i+1})$.
- We can determine all unknown syndromes using the previous theorem (majority voting) iteratively on the sequence of codes $C_L(D, G) \supset \cdots \supset C_L(D, H_i) \supset C_L(D, H_{i+1}) \supset \cdots$.
- Eventually, we then know all syndromes, after which we can determine the error-vector **e**.

Reducing complexity

It is not necessary to calculate all unknown syndromes, but one can stop the recursive computations when a code $C_L(D, H_i)$ is reached such that $n - \deg H_i - g \geq d_S(G)$.

Reducing complexity

It is not necessary to calculate all unknown syndromes, but one can stop the recursive computations when a code $C_L(D, H_i)$ is reached such that $n - \deg H_i - g \geq d_S(G)$.

Proposition

Let $\mathbf{c} \in C_L(D, G)$ *and* $S = (S_1, S_2, \dots)$ *a sequence of points not occurring in* $\operatorname{supp} D$*. Suppose that* $\mathbf{e} \in \mathbb{F}^n$ *of weight at most* $(d_S(G) - 1)/2$*. Let* $\delta = d_S(G) - n + \deg G + g$*. Suppose that we know* $s_\omega(\mathbf{e})$ *for all* $\omega \in \Omega(-D + G - S_1 - \cdots - S_\delta)$*. Then we can find* c *using the basic algorithm on the code* $C_L(D, G - S_1 - \cdots - S_\delta)$*.*

Reducing complexity

Proof:

- Write $T = S_1$ and suppose that $\mathbf{c} = \mathrm{Ev}_D(f)$ with $f \in L(G)$.
- Let $f_1, \ldots, f_k$ be a basis of $L(G)$ such that $\rho_T(f_1) < \cdots < \rho_T(f_k)$ and ω_l an element of $\Omega(-D + G - T)$ of maximal pole order at T.

Reducing complexity

Proof:

- Write $T = S_1$ and suppose that $\mathbf{c} = \mathrm{Ev}_D(f)$ with $f \in L(G)$.
- Let $f_1, \ldots, f_k$ be a basis of $L(G)$ such that $\rho_T(f_1) < \cdots < \rho_T(f_k)$ and ω_l an element of $\Omega(-D + G - T)$ of maximal pole order at T.
- We then have that any $\omega \in \Omega(-D + G - T)$ can be written as $\alpha\omega_l + \omega_r$ for certain $\omega_r \in \Omega(-D + G)$ and constant α.

Reducing complexity

Proof:

- Write $T = S_1$ and suppose that $\mathbf{c} = \mathrm{Ev}_D(f)$ with $f \in L(G)$.
- Let $f_1, \ldots, f_k$ be a basis of $L(G)$ such that $\rho_T(f_1) < \cdots < \rho_T(f_k)$ and ω_l an element of $\Omega(-D + G - T)$ of maximal pole order at T.
- We then have that any $\omega \in \Omega(-D + G - T)$ can be written as $\alpha\omega_l + \omega_r$ for certain $\omega_r \in \Omega(-D + G)$ and constant α.
- Also we can write
$$f = \sum_{i=1}^{k} \alpha_i f_i$$
and by assumption $s_{\omega_l}(\mathbf{c}) = s_{\omega_l}(\mathbf{r}) - s_{\omega_l}(\mathbf{e})$ is a known expression.

Reducing complexity

- Since $\rho_T(f_i) < \rho_T(f_k)$ for $1 \leq i < k$ and $\mathbf{c} = \mathrm{Ev}_D(f)$, we have that

$$s_{\omega_l}(\mathbf{c}) = \sum_{i=1}^{k} \alpha_i s_{\omega_l}(\mathrm{Ev}_D(f_i)) = \alpha_k s_{\omega_l}(\mathrm{Ev}_D(f_k)).$$

Reducing complexity

- Since $\rho_T(f_i) < \rho_T(f_k)$ for $1 \leq i < k$ and $\mathbf{c} = \mathrm{Ev}_D(f)$, we have that

$$s_{\omega_l}(\mathbf{c}) = \sum_{i=1}^{k} \alpha_i s_{\omega_l}(\mathrm{Ev}_D(f_i)) = \alpha_k s_{\omega_l}(\mathrm{Ev}_D(f_k)).$$

- We claim that we can always determine α_k. Indeed if $s_{\omega_m}(\mathrm{Ev}_D(f_k)) = 0$, then $s_{\omega_l}(\mathbf{c}) = 0$ implying that $\mathbf{c} \in C_L(D, G - T)$. But then $\alpha_k = 0$.

Reducing complexity

- Since $\rho_T(f_i) < \rho_T(f_k)$ for $1 \leq i < k$ and $\mathbf{c} = \mathrm{Ev}_D(f)$, we have that

$$s_{\omega_l}(\mathbf{c}) = \sum_{i=1}^{k} \alpha_i s_{\omega_l}(\mathrm{Ev}_D(f_i)) = \alpha_k s_{\omega_l}(\mathrm{Ev}_D(f_k)).$$

- We claim that we can always determine α_k. Indeed if $s_{\omega_m}(\mathrm{Ev}_D(f_k)) = 0$, then $s_{\omega_l}(\mathbf{c}) = 0$ implying that $\mathbf{c} \in C_L(D, G - T)$. But then $\alpha_k = 0$.
- If $s_{\omega_m}(\mathrm{Ev}_D(f_k)) \neq 0$, then

$$\alpha_k = \frac{s_{\omega_l}(\mathbf{c})}{s_{\omega_l}(\mathrm{Ev}_D(f_k))} = \frac{s_{\omega_l}(\mathbf{r}) - s_{\omega_l}(\mathbf{e})}{s_{\omega_l}(\mathrm{Ev}_D(f_k))}. \tag{19}$$

Reducing complexity

- We can repeat this treating $r - \alpha_k \mathrm{Ev}_D(f_k)$ as the received vector, taking $C_L(D, G - S_1)$ as the code we work with and defining $T = S_2$.
- Iterating this procedure δ times, we obtain as output a vector $r - \mathrm{Ev}_D(g)$ for an explicitly known function g such that $f - g \in L(G - S_1 - \cdots - S_\delta)$.

Reducing complexity

- We can repeat this treating $r - \alpha_k \mathrm{Ev}_D(f_k)$ as the received vector, taking $C_L(D, G - S_1)$ as the code we work with and defining $T = S_2$.
- Iterating this procedure δ times, we obtain as output a vector $r - \mathrm{Ev}_D(g)$ for an explicitly known function g such that $f - g \in L(G - S_1 - \cdots - S_\delta)$.
- The vector $r - \mathrm{Ev}_D(g)$ differs in $\mathrm{wt}(\mathbf{e}) < (n - \deg G + \delta - g)/2$ positions from $\mathrm{Ev}_D(f - g)$, so we can use the basic algorithm to find the function $f - g$ completing the decoding.

$\square$

Reducing complexity

- We can repeat this treating $r - \alpha_k \mathrm{Ev}_D(f_k)$ as the received vector, taking $C_L(D, G - S_1)$ as the code we work with and defining $T = S_2$.
- Iterating this procedure δ times, we obtain as output a vector $r - \mathrm{Ev}_D(g)$ for an explicitly known function g such that $f - g \in L(G - S_1 - \cdots - S_\delta)$.
- The vector $r - \mathrm{Ev}_D(g)$ differs in $\mathrm{wt}(\mathbf{e}) < (n - \deg G + \delta - g)/2$ positions from $\mathrm{Ev}_D(f - g)$, so we can use the basic algorithm to find the function $f - g$ completing the decoding.

☐It's time to look at an example again!

Example

Example

- Consider the curve χ given by $x_2^2 + x_2 = x_1^9$ over $\mathbb{F}_{64}$.
- It is a hyperelliptic curve of genus 4 with 129 rational points. We denote by T_∞ the unique point that has a pole at x_1 , by T_0 the point that has a zero at x_2 and by T_1 the point that has a zero at $x_2 + 1$.
- Let $G = -T_0 + 121T_\infty$ and D be the sum of the 126 rational points different from T_0, T_1 and T_∞.

Example

Example

- Consider the curve χ given by $x_2^2 + x_2 = x_1^9$ over $\mathbb{F}_{64}$.
- It is a hyperelliptic curve of genus 4 with 129 rational points. We denote by T_∞ the unique point that has a pole at x_1 , by T_0 the point that has a zero at x_2 and by T_1 the point that has a zero at $x_2 + 1$.
- Let $G = -T_0 + 121T_\infty$ and D be the sum of the 126 rational points different from T_0, T_1 and T_∞.
- The code $C_L(D, G)$ is a $[126, 117, \geq 6]$ code. We first calculate the generalized order bound for this code using the sequence $S = (T_\infty, T_\infty, \dots)$. We have that $H(T_\infty) = \langle 2, 9 \rangle$.

Example

Example

- Consider the curve χ given by $x_2^2 + x_2 = x_1^9$ over $\mathbb{F}_{64}$.
- It is a hyperelliptic curve of genus 4 with 129 rational points. We denote by T_∞ the unique point that has a pole at x_1 , by T_0 the point that has a zero at x_2 and by T_1 the point that has a zero at $x_2 + 1$.
- Let $G = -T_0 + 121T_\infty$ and D be the sum of the 126 rational points different from T_0, T_1 and T_∞.
- The code $C_L(D, G)$ is a $[126, 117, \geq 6]$ code. We first calculate the generalized order bound for this code using the sequence $S = (T_\infty, T_\infty, \dots)$. We have that $H(T_\infty) = \langle 2, 9 \rangle$.
- The differential $\omega = (x_1^{63} + 1)^{-1} dx_1$ has divisor $-D + 132T_\infty$ and can be used to show that $H(T_\infty, \mathcal{M}_S(0)) = \{i - 11 \mid i \in H(T_\infty)\} \cup \{-4\}$. We find that
$$p_{T_\infty}(t) p_{T_\infty, \mathcal{M}_S(0)}(t) = \cdots + 7t + 7t^2 + 8t^3 + 9t^4 + 10t^5 + \cdots . \quad (20)$$

Example

- The differential $\omega = (x_1^{63}+1)^{-1}dx_1$ has divisor $-D + 132T_\infty$ and can be used to show that $H(T_\infty, \mathcal{M}_S(0)) = \{i - 11 \mid i \in H(T_\infty)\} \cup \{-4\}$.

Example

- The differential $\omega = (x_1^{63}+1)^{-1}dx_1$ has divisor $-D + 132T_\infty$ and can be used to show that $H(T_\infty, \mathcal{M}_S(0)) = \{i-11 \mid i \in H(T_\infty)\} \cup \{-4\}$.
- We find that

$$p_{T_\infty}(t)p_{T_\infty,\mathcal{M}_S(0)}(t) = \cdots + 7t + 7t^2 + 8t^3 + 9t^4 + 10t^5 + \cdots .$$

This means that $d_S(G) = 7$ implying that the code we are studying is in fact a $[126, 117, \geq 7]$ code.

Example

- The differential $\omega = (x_1^{63}+1)^{-1}dx_1$ has divisor $-D + 132T_\infty$ and can be used to show that $H(T_\infty, \mathcal{M}_S(0)) = \{i - 11 \mid i \in H(T_\infty)\} \cup \{-4\}$.
- We find that

$$p_{T_\infty}(t)p_{T_\infty,\mathcal{M}_S(0)}(t) = \cdots + 7t + 7t^2 + 8t^3 + 9t^4 + 10t^5 + \cdots .$$

  This means that $d_S(G) = 7$ implying that the code we are studying is in fact a $[126, 117, \geq 7]$ code.
- We represent $\mathbb{F}_{64}$ as $\mathbb{F}_2[\gamma]$, with γ a primitive element satisfying $\gamma^6 + \gamma + 1 = 0$.

Example

- The points in $\operatorname{supp} D$ have nonzero coordinates. We write these as powers of γ with exponents between 0 and 62. Then we can order these points lexicographically after these exponents.
- In this way we get $P_1 = (1, \gamma^{21})$, ..., $P_{126} = (\gamma^{62}, \gamma^{45})$.

Example

- The points in $\operatorname{supp} D$ have nonzero coordinates. We write these as powers of γ with exponents between 0 and 62. Then we can order these points lexicographically after these exponents.
- In this way we get $P_1 = (1, \gamma^{21}), \ldots, P_{126} = (\gamma^{62}, \gamma^{45})$.
- We will need a basis $f_1, \ldots, f_{117}$ of $L(G)$ of increasing pole order in T_∞. We can take

$$f_i = \begin{cases} x_1^i & \text{if } 1 \le i \le 3, \\ x_1^{(i-5)/2} x_2 & \text{if } i \ge 5 \text{ and } i \text{ odd}, \\ x_1^{i/2} & \text{if } i \ge 4 \text{ and } i \text{ even}. \end{cases}$$

Example

Following from before we have:

i	1	2	3	4	5	6	7	8	9	10	11	12
ρ_i	0	2	4	6	8	9	10	11	12	13	14	15
h_i	1	x_1	x_1^2	x_1^3	x_1^4	x_2	x_1^5	x_1x_2	x_1^6	$x_1^2x_2$	x_1^7	$x_1^3x_2$

and (still using $\omega = (x_1^{63}+1)^{-1}dx_1$)

j	1	2	3	4	5	6	7	8	9	10	11	12
σ_j	-11	-9	-7	-5	-4	-3	-2	-1	0	1	2	3
$\frac{\omega_j}{\omega}$	1	x_1	x_1^2	x_1^3	$\frac{x_1^8}{x_2}$	x_1^4	x_2	x_1^5	x_1x_2	x_1^6	$x_1^2x_2$	x_1^7

Example

- Now define an error-vector $\mathbf{e}$ in the following way: $e_1 = 1$, $e_2 = \gamma^{42}$, $e_{93} = \gamma^{13}$, and $e_i = 0$ otherwise.
- Since $d_S(G) = 7$, we can correct this error-pattern with the majority voting algorithm. Goppa's bound for the minimum distance of the code $C_L(D, G)$ equals 6, so we need to determine $g + (7 - 6) = 5$ unknown syndromes.
- We now assume that the sent codeword was $\mathbf{c} = \mathrm{Ev}_D(\gamma x_1^{60} + x_1^{56} x_2)$, so that the received word is $\mathbf{r} = \mathbf{c} + \mathbf{e}$.
- Then we have that $\mathbf{S}^{tot}_{T\infty}(\mathbf{c})|_{14,14}$ (resp. $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})|_{14,14}$) equals ...

Example: $\mathbf{S}^{tot}_{T\infty}(\mathbf{c})|_{14,14}$

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \gamma & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 1 & \gamma & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma & 0 & 0 & 0 & 0 & 1 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\gamma & 0 & 0 & 0 & 1 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Example: $\mathbf{S}_{T_\infty}^{tot}(\mathbf{e})|_{14,14}$

$$\begin{pmatrix}
\gamma^{7} & 1 & \gamma^{45} & \gamma^{43} & \gamma^{37} & \gamma^{7} & \gamma^{54} & \gamma^{53} & \gamma^{26} & \gamma^{36} & \gamma^{30} & \gamma^{19} & \gamma^{62} & \gamma^{2} \\
1 & \gamma^{45} & \gamma^{43} & \gamma^{37} & \gamma^{54} & \gamma^{53} & \gamma^{26} & \gamma^{36} & \gamma^{30} & \gamma^{19} & \gamma^{62} & \gamma^{2} & \gamma^{46} & \gamma^{48} \\
\gamma^{45} & \gamma^{43} & \gamma^{37} & \gamma^{54} & \gamma^{26} & \gamma^{36} & \gamma^{30} & \gamma^{19} & \gamma^{62} & \gamma^{2} & \gamma^{46} & \gamma^{48} & \gamma^{16} & \gamma^{31} \\
\gamma^{43} & \gamma^{37} & \gamma^{54} & \gamma^{26} & \gamma^{30} & \gamma^{19} & \gamma^{62} & \gamma^{2} & \gamma^{46} & \gamma^{48} & \gamma^{16} & \gamma^{31} & \gamma^{13} & \gamma^{14} \\
\gamma^{5} & 0 & \gamma^{51} & \gamma^{18} & \gamma^{23} & \gamma^{62} & \gamma^{15} & \gamma^{46} & \gamma^{49} & \gamma^{16} & \gamma^{25} & \gamma^{13} & \gamma^{47} & \gamma^{36} \\
\gamma^{37} & \gamma^{54} & \gamma^{26} & \gamma^{30} & \gamma^{62} & \gamma^{2} & \gamma^{46} & \gamma^{48} & \gamma^{16} & \gamma^{31} & \gamma^{13} & \gamma^{14} & \gamma^{36} & \gamma^{60} \\
\gamma^{7} & \gamma^{53} & \gamma^{36} & \gamma^{19} & \gamma^{2} & \gamma^{50} & \gamma^{48} & \gamma^{60} & \gamma^{31} & \gamma^{28} & \gamma^{14} & \gamma^{3} & \gamma^{60} & \gamma^{61} \\
\gamma^{54} & \gamma^{26} & \gamma^{30} & \gamma^{62} & \gamma^{46} & \gamma^{48} & \gamma^{16} & \gamma^{31} & \gamma^{13} & \gamma^{14} & \gamma^{36} & \gamma^{60} & \gamma^{22} & \gamma^{43} \\
\gamma^{53} & \gamma^{36} & \gamma^{19} & \gamma^{2} & \gamma^{48} & \gamma^{60} & \gamma^{31} & \gamma^{28} & \gamma^{14} & \gamma^{3} & \gamma^{60} & \gamma^{61} & \gamma^{43} & \gamma^{7} \\
\gamma^{26} & \gamma^{30} & \gamma^{62} & \gamma^{46} & \gamma^{16} & \gamma^{31} & \gamma^{13} & \gamma^{14} & \gamma^{36} & \gamma^{60} & \gamma^{22} & \gamma^{43} & \gamma^{35} & \gamma^{26} \\
\gamma^{36} & \gamma^{19} & \gamma^{2} & \gamma^{48} & \gamma^{31} & \gamma^{28} & \gamma^{14} & \gamma^{3} & \gamma^{60} & \gamma^{61} & \gamma^{43} & \gamma^{7} & \gamma^{26} & 1 \\
\gamma^{30} & \gamma^{62} & \gamma^{46} & \gamma^{16} & \gamma^{13} & \gamma^{14} & \gamma^{36} & \gamma^{60} & \gamma^{22} & \gamma^{43} & \gamma^{35} & \gamma^{26} & \gamma^{34} & \gamma^{9} \\
\gamma^{19} & \gamma^{2} & \gamma^{48} & \gamma^{31} & \gamma^{14} & \gamma^{3} & \gamma^{60} & \gamma^{61} & \gamma^{43} & \gamma^{7} & \gamma^{26} & 1 & \gamma^{9} & \gamma^{45} \\
\gamma^{62} & \gamma^{46} & \gamma^{16} & \gamma^{13} & \gamma^{36} & \gamma^{60} & \gamma^{22} & \gamma^{43} & \gamma^{35} & \gamma^{26} & \gamma^{34} & \gamma^{9} & \gamma^{48} & \gamma^{55}
\end{pmatrix}$$

Example

- In the decoding algorithm, we know the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})|_{14,14}$, which is the sum of the two previous matrices. The individual matrices are unknown to the receiver.
- Note that $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ and $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$ are guaranteed to be the same in all those positions (i, j) satisfying $\sigma_i + \rho_j \leq 0$, since these positions contain the known syndromes.

Example

- In the decoding algorithm, we know the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})|_{14,14}$, which is the sum of the two previous matrices. The individual matrices are unknown to the receiver.
- Note that $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ and $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$ are guaranteed to be the same in all those positions (i,j) satisfying $\sigma_i + \rho_j \leq 0$, since these positions contain the known syndromes.
- We now calculate $f = \gamma x_1^{60} + x_1^{56} x_2$. Since $f \in L(G)$, we can write $f = \sum_{i=1}^{117} \alpha_i f_i$. We will determine α_{113} up till α_{117} using majority voting.

Example

- In the decoding algorithm, we know the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})|_{14,14}$, which is the sum of the two previous matrices. The individual matrices are unknown to the receiver.
- Note that $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ and $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$ are guaranteed to be the same in all those positions (i,j) satisfying $\sigma_i + \rho_j \leq 0$, since these positions contain the known syndromes.
- We now calculate $f = \gamma x_1^{60} + x_1^{56} x_2$. Since $f \in L(G)$, we can write $f = \sum_{i=1}^{117} \alpha_i f_i$. We will determine α_{113} up till α_{117} using majority voting.
- In the first step of the algorithm we need to determine which positions (i,j) satisfying $\sigma_i + \rho_j = 1$, are candidates as well.

Example

- In the decoding algorithm, we know the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})|_{14,14}$, which is the sum of the two previous matrices. The individual matrices are unknown to the receiver.
- Note that $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ and $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$ are guaranteed to be the same in all those positions (i, j) satisfying $\sigma_i + \rho_j \leq 0$, since these positions contain the known syndromes.
- We now calculate $f = \gamma x_1^{60} + x_1^{56} x_2$. Since $f \in L(G)$, we can write $f = \sum_{i=1}^{117} \alpha_i f_i$. We will determine α_{113} up till α_{117} using majority voting.
- In the first step of the algorithm we need to determine which positions (i, j) satisfying $\sigma_i + \rho_j = 1$, are candidates as well.
- From the series expansion of $p_{T_\infty}(t) p_{T_\infty, \mathcal{M}_S(0)}(t)$ we get that there are at most 7 such positions (i, j).

Example: Decoding

- By row reduction of the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ we get that $(1,1)$ and $(2,2)$ are the only discrepancies in the known part $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$.

Example: Decoding

- By row reduction of the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ we get that $(1,1)$ and $(2,2)$ are the only discrepancies in the known part $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$.
- The candidates in the first and following steps can therefore not contain a 1 or a 2 in any of their coordinates.

Example: Decoding

- By row reduction of the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ we get that $(1,1)$ and $(2,2)$ are the only discrepancies in the known part $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$.
- The candidates in the first and following steps can therefore not contain a 1 or a 2 in any of their coordinates.
- The votes can be calculated directly once the candidates are known. The results of the first step of the algorithm is:

candidate	$(6,3)$	$(4,4)$	$(3,5)$
vote	γ^{26}	γ^{26}	γ^{26}

- We conclude that $s_{\omega_{10}}(\mathbf{e}) = \gamma^{26}$. Using the equation, we get $\alpha_{117} = 1$, and we can then replace $\mathbf{S}^{tot}_{T\infty}(\mathbf{r})$ by the matrix $\mathbf{S}^{tot}_{T\infty}(\mathbf{r} - \mathrm{Ev}_D(f_{117}))$.

Example: Decoding

- Since the voting is unanimous, there are no new discrepancies.

Example: Decoding

- Since the voting is unanimous, there are no new discrepancies.
- In the second step of the algorithm, we get:

candidate	$(7,3)$	$(5,4)$	$(3,6)$
vote	γ^{36}	γ^{36}	γ^{36}

- Therefore $s_{\omega_{10}}(\mathbf{e}) = \gamma^{36}$ and $\alpha_{116} = \gamma$. In this particular example the updated syndrome matrix now becomes $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$, because of our choice of the sent codeword $\mathbf{c}$.

Example: Decoding

- Since the voting is unanimous, there are no new discrepancies.
- In the second step of the algorithm, we get:

candidate	$(7,3)$	$(5,4)$	$(3,6)$
vote	γ^{36}	γ^{36}	γ^{36}

- Therefore $s_{\omega_{10}}(\mathbf{e}) = \gamma^{36}$ and $\alpha_{116} = \gamma$. In this particular example the updated syndrome matrix now becomes $\mathbf{S}^{tot}_{T\infty}(\mathbf{e})$, because of our choice of the sent codeword $\mathbf{c}$.
- Continuing to the third step, we find:

candidate	$(8,3)$	$(6,4)$	$(4,5)$	$(3,7)$
vote	γ^{30}	γ^{30}	γ^{30}	γ^{30}

- Thus $s_{\omega_{11}}(\mathbf{e}) = \gamma^{30}$ and $\alpha_{115} = 0$.

Example: Decoding

- The fourth step yields:

candidate	$(9,3)$	$(7,4)$	$(5,5)$	$(4,6)$	$(3,8)$
vote	γ^{19}	γ^{19}	γ^{19}	γ^{19}	γ^{19}

This implies that $s_{\omega_{12}}(\mathbf{e}) = \gamma^{19}$ and $\alpha_{114} = 0$.

Example: Decoding

- The fourth step yields:

candidate	$(9,3)$	$(7,4)$	$(5,5)$	$(4,6)$	$(3,8)$
vote	γ^{19}	γ^{19}	γ^{19}	γ^{19}	γ^{19}

  This implies that $s_{\omega_{12}}(\mathbf{e}) = \gamma^{19}$ and $\alpha_{114} = 0$.

- The fifth and last step gives:

candidate	$(10,3)$	$(8,4)$	$(6,5)$	$(5,6)$	$(4,7)$	$(3,9)$
vote	γ^{62}	γ^{62}	γ^{62}	γ^{49}	γ^{62}	γ^{62}

- In this case the voting is not unanimous and we find $s_{\omega_{13}}(\mathbf{e}) = \gamma^{62}$ and $\alpha_{113} = 0$.
- The reason the voting is not unanimous in this case, is that the $(5,6)$-th position is a discrepancy in the matrix of syndromes.

Contents

List decoding

- We will describe a *list decoding* algorithm for algebraic geometry codes. This is is an extension of the basic algorithm.
- Suppose we use the code $C_L(D, G)$ and that we have received $(r_1, \ldots, r_n)$ containing at most τ errors.

List decoding

- We will describe a *list decoding* algorithm for algebraic geometry codes. This is is an extension of the basic algorithm.
- Suppose we use the code $C_L(D, G)$ and that we have received $(r_1, \ldots, r_n)$ containing at most τ errors.
- The algorithm works with:
  - A divisor A with $\operatorname{supp} A \cap \operatorname{supp} D = \varnothing$ satisfying certain conditions to be described
  - A natural number s known as the multiplicty parameter.

List decoding

- We will describe a *list decoding* algorithm for algebraic geometry codes. This is is an extension of the basic algorithm.
- Suppose we use the code $C_L(D, G)$ and that we have received $(r_1, \ldots, r_n)$ containing at most τ errors.
- The algorithm works with:
  - A divisor A with $\mathrm{supp}\, A \cap \mathrm{supp}\, D = \varnothing$ satisfying certain conditions to be described
  - A natural number s known as the multiplicty parameter.

The idea: Find a nonzero polynomial $Q(y) \in \mathscr{F}[y]$ such that:

(i) $Q(y) = Q_0 + \cdots + Q_\lambda y^\lambda$ where $Q_i \in L(A - iG), i = 0, \ldots, \lambda$

(ii) $Q(y)$ has a zero of multiplicity s in (P_j, r_j), $j = 1, \ldots, n$

List decoding as extension of the basic algorithm

- The multiplicty conditions in (ii) means: Let t be a local parameter at P_j and $Q(y) = \sum \mu_{a,b} t^a (y - r_j)^b$, then $\mu_{a,b} = 0$ for all $a + b < s$

List decoding as extension of the basic algorithm

- The multiplicty conditions in (ii) means: Let t be a local parameter at P_j and $Q(y) = \sum \mu_{a,b} t^a (y - r_j)^b$, then $\mu_{a,b} = 0$ for all $a + b < s$
- This is an extension of the basic algorithm in two ways.
  - Larger y-degree of Q is allowed.
  - Larger multiplicity of the zeroes of Q is allowed.
- In this way, as we shall see, we are able to correct a larger number of errors if we accept a list of possible codewords.

List decoding as extension of the basic algorithm

- The multiplicty conditions in (ii) means: Let t be a local parameter at P_j and $Q(y) = \sum \mu_{a,b} t^a (y - r_j)^b$, then $\mu_{a,b} = 0$ for all $a + b < s$
- This is an extension of the basic algorithm in two ways.
  - Larger y-degree of Q is allowed.
  - Larger multiplicity of the zeroes of Q is allowed.
- In this way, as we shall see, we are able to correct a larger number of errors if we accept a list of possible codewords.
- The conditions on the divisor A are as follows.
  1. $\deg A < s(n - \tau)$
  2. $\deg A > \frac{ns(s+1)}{2(\lambda+1)} + \frac{\lambda \deg G}{2} + g - 1$

  It can be seen that if $\tau < n - \frac{n(s+1)}{2(\lambda+1)} - \frac{\lambda \deg G}{2s} - \frac{g}{s}$ then such a divisor A exists.

List decoding: Basic lemma

Lemma

Suppose the transmitted word is generated by $f \in L(G)$ *and* $Q(y)$ *satisfies* (i) *and* (ii) *then* $Q(f) = 0$

List decoding: Basic lemma

Lemma

Suppose the transmitted word is generated by $f \in L(G)$ and $Q(y)$ satisfies (i) and (ii) then $Q(f) = 0$

Proof:

- Since $f \in L(G)$ and $Q_i \in L(A - iG)$ we have $f^i Q_i \in L(A)$ and therefore $Q(f) \in L(A)$.

List decoding: Basic lemma

Lemma

Suppose the transmitted word is generated by $f \in L(G)$ and $Q(y)$ satisfies (i) and (ii) then $Q(f) = 0$

Proof:

- Since $f \in L(G)$ and $Q_i \in L(A - iG)$ we have $f^i Q_i \in L(A)$ and therefore $Q(f) \in L(A)$.
- $Q(f(P_j))$ has a zero of multiplicity s in P_j for at least $n - \tau$ j's $\in \{1, 2, \ldots, n\}$ so that $Q(f) \in L(A - sP_{i_1} - \cdots - sP_{i_r})$ with $r \geq n - \tau$.
- But $\deg(A - sP_{i_1} - \cdots - sP_{i_r}) < 0$ and therefore $Q(f) = 0$.

List decoding: Basic lemma

Lemma

Suppose the transmitted word is generated by $f \in L(G)$ and $Q(y)$ satisfies (i) and (ii) then $Q(f) = 0$

Proof:

- Since $f \in L(G)$ and $Q_i \in L(A - iG)$ we have $f^i Q_i \in L(A)$ and therefore $Q(f) \in L(A)$.
- $Q(f(P_j))$ has a zero of multiplicity s in P_j for at least $n - \tau$ j's $\in \{1, 2, \ldots, n\}$ so that $Q(f) \in L(A - sP_{i_1} - \cdots - sP_{i_r})$ with $r \geq n - \tau$.
- But $\deg(A - sP_{i_1} - \cdots - sP_{i_r}) < 0$ and therefore $Q(f) = 0$.
- Thus if the divisor A satisfies condition (1), then the function f gives a factor $y - f$ in $Q(y)$.

List decoding: Existence of $Q(y)$

- Later we will discuss how such factors are actually found.
- Now we show the existence of the interpolation polynomial Q.

Lemma

If $\deg A$ *satisfies (2) above then a nonzero* $Q(y) \in \mathscr{F}[y]$ *satisfying (i) and (ii) exists.*

List decoding: Existence of $Q(y)$

- Later we will discuss how such factors are actually found.
- Now we show the existence of the interpolation polynomial Q.

Lemma

If $\deg A$ *satisfies (2) above then a nonzero* $Q(y) \in \mathscr{F}[y]$ *satisfying (i) and (ii) exists.*

Proof:
By selecting bases for the spaces $L(A - iG), i = 0, 1, \ldots, \lambda$ the condition (ii) translates into a system of homogeneous linear equations in $\sum_{i=0}^{\lambda} l(A - iG)$ unknowns. The number of equations is $\frac{n(s+1)s}{2}$ which by (2) is smaller than the number of unknowns, so there is a nonzero solution to the system. □

Algorithm

This leads to the following algorithm:

Input: A received word $r = (r_1, r_2, \ldots, r_n)$.
Find a polynomial $Q(y)$ satisfying (i) and (ii).
Find factors of $Q(y)$ of the form $y - f$ with $f \in L(G)$.
If no such factors exist **Output**: Failure.
Else **Output** : $\mathrm{Ev}_D(f)$ for those f's where $d(\mathrm{Ev}_D(f), r) \leq \tau$.

Algorithm

This leads to the following algorithm:

Input: A received word $r = (r_1, r_2, \ldots, r_n)$.
Find a polynomial $Q(y)$ satisfying (i) and (ii).
Find factors of $Q(y)$ of the form $y - f$ with $f \in L(G)$.
If no such factors exist **Output**: Failure.
Else **Output** : $\mathrm{Ev}_D(f)$ for those f's where $d(\mathrm{Ev}_D(f), r) \leq \tau$.

- It can be seen that this algorithm only improves on $\frac{n - \deg G}{2}$ if $\lambda \geq s$ and
$$n\left(1 - \frac{s+1}{\lambda+1}\right) > \left(\frac{\lambda}{s} - 1\right) \deg G + \frac{2g}{s} + 1$$
- For fixed λ the optimal s is
$$\left\lfloor \left[\frac{2(\lambda+1)}{n}\left(\frac{\lambda}{2} \deg G + g\right)\right]^{\frac{1}{2}} \right\rfloor$$

Example

Example:
In a previous example we used a $[60, 18, \geq 37]$ code over $\mathbb{F}_{16}$.

Example

Example:

In a previous example we used a $[60, 18, \geq 37]$ code over $\mathbb{F}_{16}$.

- With $\lambda = 6$ and $s = 4$ we can correct 19 errors using list decoding.
- With $\lambda = 10$ and $s = 7$, 20 errors can be corrected
- With $\lambda = 50$ and $s = 32$, 22 errors can be corrected.

□

Example

Example:

In a previous example we used a $[60, 18, \geq 37]$ code over $\mathbb{F}_{16}$.

- With $\lambda = 6$ and $s = 4$ we can correct 19 errors using list decoding.
- With $\lambda = 10$ and $s = 7$, 20 errors can be corrected
- With $\lambda = 50$ and $s = 32$, 22 errors can be corrected.

□As we have seen, and we will discuss this further in the next section, the polynomial $Q(y)$ can be found by solving a system of homogenous linear equations.

Finding factors of $Q(y)$

- We will address the question of finding the relevant factors of the polynomial $Q(y)$ and present two different methods for doing that.

Finding factors of $Q(y)$

- We will address the question of finding the relevant factors of the polynomial $Q(y)$ and present two different methods for doing that.
- The first method transforms the problem to that of finding factors of a univariate polynomial over a large finite field, and the second one uses Hensel lifting.

Finding factors of $Q(y)$

- We will address the question of finding the relevant factors of the polynomial $Q(y)$ and present two different methods for doing that.
- The first method transforms the problem to that of finding factors of a univariate polynomial over a large finite field, and the second one uses Hensel lifting.
- The first algorithm reduces the problem of finding factors of the form $y - f$ in $Q(y)$, to the problem of finding roots of a polynomial $\widehat{Q}(y)$ in $\mathbb{F}_{q^m}$ obtained by "reducing" the coefficients of $Q(y)$ modulo a point R of sufficiently large degree m where $R \notin \mathrm{supp}\, A$ and $R \notin \mathrm{supp}\, G$.

Finding factors of $Q(y)$

- We will address the question of finding the relevant factors of the polynomial $Q(y)$ and present two different methods for doing that.
- The first method transforms the problem to that of finding factors of a univariate polynomial over a large finite field, and the second one uses Hensel lifting.
- The first algorithm reduces the problem of finding factors of the form $y - f$ in $Q(y)$, to the problem of finding roots of a polynomial $\widehat{Q}(y)$ in $\mathbb{F}_{q^m}$ obtained by "reducing" the coefficients of $Q(y)$ modulo a point R of sufficiently large degree m where $R \notin \operatorname{supp} A$ and $R \notin \operatorname{supp} G$.
- It can be seen that such a point exists. The reduction is performed by evaluating the functions Q_i in R.

Finding factors of $Q(y)$

- One then finds zeroes of $\widehat{Q}(y)$ using a root-finding algorithm for finite fields and for those zeroes that lie in $\mathrm{Ev}_R(L(G))$ one finds the corresponding f's $\in L(G)$.

Finding factors of $Q(y)$

- One then finds zeroes of $\widehat{Q}(y)$ using a root-finding algorithm for finite fields and for those zeroes that lie in $\mathrm{Ev}_R(L(G))$ one finds the corresponding f's $\in L(G)$.
- For this to be possible the map $\mathrm{Ev}_R : L(G) \to \mathbb{F}_{q^m}$ shall be injective and this is the case if $\deg R > \deg G$.

Finding factors of $Q(y)$

- One then finds zeroes of $\widehat{Q}(y)$ using a root-finding algorithm for finite fields and for those zeroes that lie in $\mathrm{Ev}_R(L(G))$ one finds the corresponding f's $\in L(G)$.
- For this to be possible the map $\mathrm{Ev}_R : L(G) \to \mathbb{F}_{q^m}$ shall be injective and this is the case if $\deg R > \deg G$.
- We need a way to evaluate functions from $L(G)$ and $L(A - iG)$ in R, and also a method for reconstructing an f from an element in $\mathrm{Ev}_R(L(G)) \subseteq \mathbb{F}_{q^m}$.
- We shall now assume w.l.o.g that the divisor G is effective and also that $A \geq G$. This implies that $L(G) \subseteq L(A)$ and also that $L(A - iG) \subseteq L(A)$.

Finding factors of $Q(y)$ as roots of $\widehat{Q}(y)$

- Let ϕ_1, ϕ_2, ..., ϕ_k be a basis of $L(G)$ (as a $\mathbb{F}_q$–vector space).
- Let ϕ_1, ..., ϕ_k, ϕ_{k+1}, ..., ϕ_a be a basis of $L(A)$.
- R can the be "represented" by the values $\phi_1(R), \phi_2(R), \ldots, \phi_a(R)$ i.e. an element of $\mathbb{F}_{q^m}^a$.

Finding factors of $Q(y)$ as roots of $\widehat{Q}(y)$

- Let ϕ_1, ϕ_2, ..., ϕ_k be a basis of $L(G)$ (as a $\mathbb{F}_q$–vector space).
- Let ϕ_1, ..., ϕ_k, ϕ_{k+1}, ..., ϕ_a be a basis of $L(A)$.
- R can the be "represented" by the values $\phi_1(R), \phi_2(R), \ldots, \phi_a(R)$ i.e. an element of $\mathbb{F}_{q^m}^a$.
- Let $Q_i = \sum_{j=1}^a \gamma_{i,j}\phi_j$ then $Q(y) = \sum_{i=0}^\lambda \sum_{j=1}^a \gamma_{i,j}\phi_j y^i$ and $\widehat{Q}(y) = \sum_{i=0}^\lambda \sum_{j=1}^a \gamma_{i,j}\phi_j(R)y^i$.

Finding factors of $Q(y)$ as roots of $\widehat{Q}(y)$

- Let ϕ_1, ϕ_2, …, ϕ_k be a basis of $L(G)$ (as a $\mathbb{F}_q$–vector space).
- Let ϕ_1, …, ϕ_k, ϕ_{k+1}, …, ϕ_a be a basis of $L(A)$.
- R can the be "represented" by the values $\phi_1(R), \phi_2(R), \ldots, \phi_a(R)$ i.e. an element of $\mathbb{F}_{q^m}^a$.
- Let $Q_i = \sum_{j=1}^{a} \gamma_{i,j}\phi_j$ then $Q(y) = \sum_{i=0}^{\lambda} \sum_{j=1}^{a} \gamma_{i,j}\phi_j y^i$ and $\widehat{Q}(y) = \sum_{i=0}^{\lambda} \sum_{j=1}^{a} \gamma_{i,j}\phi_j(R) y^i$.
- If $\beta \in \mathbb{F}_{q^m}$ is a zero of $\widehat{Q}(y)$ we shall then find $(f_1, f_2, \ldots, f_k) \in \mathbb{F}_q$ such that $\sum_{l=1}^{k} f_l \phi_l(R) = \beta$.

Finding factors of $Q(y)$ as roots of $\widehat{Q}(y)$

- Let $\phi_1, \phi_2, \ldots, \phi_k$ be a basis of $L(G)$ (as a $\mathbb{F}_q$–vector space).
- Let $\phi_1, \ldots, \phi_k, \phi_{k+1}, \ldots, \phi_a$ be a basis of $L(A)$.
- R can the be "represented" by the values $\phi_1(R), \phi_2(R), \ldots, \phi_a(R)$ i.e. an element of $\mathbb{F}_{q^m}^a$.
- Let $Q_i = \sum_{j=1}^{a} \gamma_{i,j}\phi_j$ then $Q(y) = \sum_{i=0}^{\lambda} \sum_{j=1}^{a} \gamma_{i,j}\phi_j y^i$ and $\widehat{Q}(y) = \sum_{i=0}^{\lambda} \sum_{j=1}^{a} \gamma_{i,j}\phi_j(R) y^i$.
- If $\beta \in \mathbb{F}_{q^m}$ is a zero of $\widehat{Q}(y)$ we shall then find $(f_1, f_2, \ldots, f_k) \in \mathbb{F}_q$ such that $\sum_{l=1}^{k} f_l\phi_l(R) = \beta$.
- Using a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ this gives m linear equations in k unknowns and there are either none or a unique solution.
- In the latter case we have found an f and if $d(\mathrm{Ev}_D(f), r) \leq \tau$ we put $\mathrm{Ev}_D(f)$ on the list.

Finding factors of $Q(y)$ using Hensel lifting

In the second algorithm the idea is the following:

Finding factors of $Q(y)$ using Hensel lifting

In the second algorithm the idea is the following:

- Let P be a point, $P \notin \operatorname{supp} A$ and $P \notin \operatorname{supp} G$ and let t be a local parameter at P. Then a function in $L(G)$ can be developed as a power series in t, $f = \sum_{i=0}^{\infty} a_i t^i$.

Finding factors of $Q(y)$ using Hensel lifting

In the second algorithm the idea is the following:

- Let P be a point, $P \notin \operatorname{supp} A$ and $P \notin \operatorname{supp} G$ and let t be a local parameter at P. Then a function in $L(G)$ can be developed as a power series in t, $f = \sum_{i=0}^{\infty} a_i t^i$.
- The polynomial $Q(y)$ can also be considered as element of $\mathbb{F}_q[[t]][y]$, $Q(y) = Q_0(t, y) = \sum_{i=0,j=0}^{\infty,\lambda} \alpha_{i,j} t^i y^j$, so if $Q(f) = 0$ we get

$$Q_0(t, \sum_{i=0}^{\infty} a_i t^i) = 0 \tag{21}$$

Finding factors of $Q(y)$ using Hensel lifting

In the second algorithm the idea is the following:

- Let P be a point, $P \notin \operatorname{supp} A$ and $P \notin \operatorname{supp} G$ and let t be a local parameter at P. Then a function in $L(G)$ can be developed as a power series in t, $f = \sum_{i=0}^{\infty} a_i t^i$.
- The polynomial $Q(y)$ can also be considered as element of $\mathbb{F}_q[[t]][y]$, $Q(y) = Q_0(t, y) = \sum_{i=0,j=0}^{\infty,\lambda} \alpha_{i,j} t^i y^j$, so if $Q(f) = 0$ we get

$$Q_0(t, \sum_{i=0}^{\infty} a_i t^i) = 0 \tag{21}$$

- If we consider this equation modulo increasing powers of t it is possible to determine the a_i's recursively.

Finding factors of $Q(y)$ using Hensel lifting

- In the first step we look at equation (21) mod t which is the same as $Q_0(0, a_0) = 0$ and this is

$$\sum_{j=0}^{\lambda} \alpha_{0,j} {a_0}^j = 0 \tag{22}$$

Finding factors of $Q(y)$ using Hensel lifting

- In the first step we look at equation (21) mod t which is the same as $Q_0(0, a_0) = 0$ and this is

$$\sum_{j=0}^{\lambda} \alpha_{0,j} {a_0}^j = 0 \tag{22}$$

- Here we can suppose that $\alpha_{0,j} \neq 0$ for some j since if not $Q_0(t, y) = tR(t, y)$ and we would get $R(t, f) = 0$.
- This means that we can determine a_0 as a zero in $\mathbb{F}_q$ of the polynomial $Q_0(0, T)$.

Finding factors of $Q(y)$ using Hensel lifting

- In the first step we look at equation (21) mod t which is the same as $Q_0(0, a_0) = 0$ and this is

$$\sum_{j=0}^{\lambda} \alpha_{0,j} {a_0}^j = 0 \tag{22}$$

- Here we can suppose that $\alpha_{0,j} \neq 0$ for some j since if not $Q_0(t, y) = tR(t, y)$ and we would get $R(t, f) = 0$.
- This means that we can determine a_0 as a zero in $\mathbb{F}_q$ of the polynomial $Q_0(0, T)$.
- To determine the remaining coefficients a_i, we let for $i \geq 0$, $\psi_i(t) = \sum_{s=i}^{\infty} a_s t^{s-i}$, $M_i(t, y) = t^{-r_i} Q_i(t, y)$ where r_i is the largest integer such that t^{r_i} divides $Q_i(t, ty + a_i)$.

Finding factors of $Q(y)$ using Hensel lifting

- We then "update" the interpolation polynomial by
$$Q_{i+1}(t,y) = M_i(t, ty + a_i).$$
- Note that $Q_{i+1}(t,y)$ and r_i may depend on the value found for a_i in the previous step of the algorithm, but for simplicity we suppress this in the notation.

Finding factors of $Q(y)$ using Hensel lifting

- We then "update" the interpolation polynomial by
$$Q_{i+1}(t, y) = M_i(t, ty + a_i).$$
- Note that $Q_{i+1}(t, y)$ and r_i may depend on the value found for a_i in the previous step of the algorithm, but for simplicity we suppress this in the notation.

Lemma

$Q_i(t, \psi_i(t)) = 0$, $M_i(0, a_i) = 0$ and $M_i(0, y) \neq 0$.

Finding factors of $Q(y)$ using Hensel lifting

- We then "update" the interpolation polynomial by
$$Q_{i+1}(t,y) = M_i(t, ty + a_i).$$
- Note that $Q_{i+1}(t,y)$ and r_i may depend on the value found for a_i in the previous step of the algorithm, but for simplicity we suppress this in the notation.

Lemma

$Q_i(t,\psi_i(t)) = 0$, $M_i(0,a_i) = 0$ and $M_i(0,y) \neq 0$.

Proof:

- The y–degrees of $Q_i(t,y)$ are the same for all i and that $Q_i(t,y) \neq 0$ so r_i is well-defined.

Finding factors of $Q(y)$ using Hensel lifting

- We then "update" the interpolation polynomial by
$$Q_{i+1}(t, y) = M_i(t, ty + a_i).$$
- Note that $Q_{i+1}(t, y)$ and r_i may depend on the value found for a_i in the previous step of the algorithm, but for simplicity we suppress this in the notation.

Lemma

$Q_i(t, \psi_i(t)) = 0$, $M_i(0, a_i) = 0$ and $M_i(0, y) \neq 0$.

Proof:

- The y–degrees of $Q_i(t, y)$ are the same for all i and that $Q_i(t, y) \neq 0$ so r_i is well-defined.
- Since t does not divide $M_i(t, y)$ we have $M_i(0, y) \neq 0$.

Finding factors of $Q(y)$ using Hensel lifting

- We can now prove that $Q_i(t, \psi_i(t)) = 0$ by induction on i. The basis $i = 0$ follows by definition.

Finding factors of $Q(y)$ using Hensel lifting

- We can now prove that $Q_i(t, \psi_i(t)) = 0$ by induction on i. The basis $i = 0$ follows by definition.
- For the induction step if $Q_i(t, \psi_i(t)) = 0$ then $\psi_{i+1}(t) = (\psi_i(t) - a_i)/t$ is a y-root of $Q_i(t, ty + a_i)$ and hence of $Q_{i+1}(t, y) = t^{-r_i} Q_i(t, ty + a_i)$. By substituting $t = 0$ in $M_i(t, \psi_i(t)) = t^{-r_i} Q_i(t, \psi_i(t)) = 0$ we obtain $M_i(0, a_i) = 0$.

Finding factors of $Q(y)$ using Hensel lifting

- We can now prove that $Q_i(t, \psi_i(t)) = 0$ by induction on i. The basis $i = 0$ follows by definition.
- For the induction step if $Q_i(t, \psi_i(t)) = 0$ then $\psi_{i+1}(t) = (\psi_i(t) - a_i)/t$ is a y-root of $Q_i(t, ty + a_i)$ and hence of $Q_{i+1}(t, y) = t^{-r_i} Q_i(t, ty + a_i)$. By substituting $t = 0$ in $M_i(t, \psi_i(t)) = t^{-r_i} Q_i(t, \psi_i(t)) = 0$ we obtain $M_i(0, a_i) = 0$.

□

- The coefficients a_i can be found by solving an equation of degree λ.
- In fact the total number of solutions f is at most λ, as can be seen from the following lemma ...

Finding factors of $Q(y)$ using Hensel lifting

Lemma

Let $M_1(t,y) = \sum_{j=0}^{\lambda} M^{(j)}(t)y^j$ *be a nonzero polynomial in* $\mathbb{F}_q[[t]][y]$ *and let* β *be zero of* $M_1(0,y)$ *of multiplicity* m_β*. Define*

$$M_2(t,y) = t^{-r} M_1(t, ty + \beta),$$

where r *is the largest integer such that* t^r *divides* $M_1(t, ty + \beta)$ *then* $\deg_y M_2(0,y) \leq m_\beta$*.*

Finding factors of $Q(y)$ using Hensel lifting

Lemma

Let $M_1(t,y) = \sum_{j=0}^{\lambda} M^{(j)}(t)y^j$ *be a nonzero polynomial in* $\mathbb{F}_q[[t]][y]$ *and let* β *be zero of* $M_1(0,y)$ *of multiplicity* m_β*. Define*

$$M_2(t,y) = t^{-r}M_1(t,ty+\beta),$$

where r *is the largest integer such that* t^r *divides* $M_1(t,ty+\beta)$ *then* $\deg_y M_2(0,y) \leq m_\beta$*.*

Proof:

- Let $\widehat{M}(t,y) = M_1(t,y+\beta) = \sum_{j=0}^{\lambda} q_j(t)y^j$ then $q_j(0) = 0$ for $0 \leq j < m_\beta$ and $q_{m_\beta}(0) \neq 0$.
- Equivalently t divides $q_j(t)$ for $0 \leq j < m_\beta$ but it does not divide $q_{m_\beta}(0)$.

Finding factors of $Q(y)$ using Hensel lifting

- This means that t divides $\widehat{M}(t, ty)$ but $t^{m_\beta+1}$ does not, so $r \leq m_\beta$.

Finding factors of $Q(y)$ using Hensel lifting

- This means that t divides $\widehat{M}(t, ty)$ but $t^{m_\beta+1}$ does not, so $r \leq m_\beta$.
- Since $M_2(t, y) = t^{-r}M_1(t, ty + \beta) = \sum_{j=m_\beta}^{\lambda} q_j(t)t^{j-r}y^j$ we get $M_2(0, y) = \sum_{j=m_\beta}^{\lambda}(q_j(t)t^{j-r})|_{t=0}y^j$.
- So $\deg_y M_2(0, y) \leq r \leq m_\beta$. □

Corollary

The number of different f's is at most λ.

Finding factors of $Q(y)$ using Hensel lifting

- This means that t divides $\widehat{M}(t, ty)$ but $t^{m_\beta+1}$ does not, so $r \le m_\beta$.
- Since $M_2(t, y) = t^{-r} M_1(t, ty + \beta) = \sum_{j=m_\beta}^{\lambda} q_j(t) t^{j-r} y^j$ we get $M_2(0, y) = \sum_{j=m_\beta}^{\lambda} (q_j(t) t^{j-r})|_{t=0} y^j$.
- So $\deg_y M_2(0, y) \le r \le m_\beta$. □

Corollary

The number of different f's is at most λ.

Proof:

- Denote by A_i the set of all solutions $\mathbf{a} = (a_0, \ldots, a_i)$ the algorithm finds after i steps.
- We will show by induction that

$$\sum_{\mathbf{a} \in A_i} m_{a_i} \le \lambda. \tag{23}$$

Finding factors of $Q(y)$ using Hensel lifting

- This will imply the corollary, since then $\#A_i \leq \lambda$ for all i.

Finding factors of $Q(y)$ using Hensel lifting

- This will imply the corollary, since then $\#A_i \leq \lambda$ for all i.
- For $i = 0$ equation (23) is true, since all found a_0's in the start of the algorithm are roots of $Q_0(0, y)$ and $\deg_y Q_0(0, y) = \lambda$.

Finding factors of $Q(y)$ using Hensel lifting

- This will imply the corollary, since then $\#A_i \leq \lambda$ for all i.
- For $i = 0$ equation (23) is true, since all found a_0's in the start of the algorithm are roots of $Q_0(0, y)$ and $\deg_y Q_0(0, y) = \lambda$.
- Now suppose the result is true for i. Given a fixed $(a_0, \dots, a_i)$ at this stage of the algorithm, the a_{i+1}'s the algorithm finds in the next step are, according to the lemma, roots of a polynomial of degree at most m_{a_i} so the sum of their multiplicities is at most m_{a_i}.
- This implies that $\sum_{\mathbf{a} \in A_{i+1}} m_{a_{i+1}} \leq \sum_{\mathbf{a} \in A_i} m_{a_i} \leq \lambda$. $\square$

Example

- The only remaining issue is to bound the number of a_i's we have to determine in order to reconstruct the function $f \in L(G)$.
- To this end let $k = \dim L(G)$and let $b_1, b_2, \ldots, b_k$ be a basis of $L(G)$ such that $j_i = v_P(b_i) < v_P(b_{i+1}) = j_{i+1}$, $i = 1, \ldots, k-1$.
- This means that f is determined if we know the a_i's up to $i = j_k$. Since $b_k \in L(G - j_k P)$ we have $j_k \leq \deg G$.

Example

- The only remaining issue is to bound the number of a_i's we have to determine in order to reconstruct the function $f \in L(G)$.
- To this end let $k = \dim L(G)$and let $b_1, b_2, \ldots, b_k$ be a basis of $L(G)$ such that $j_i = v_P(b_i) < v_P(b_{i+1}) = j_{i+1}$, $i = 1, \ldots, k-1$.
- This means that f is determined if we know the a_i's up to $i = j_k$. Since $b_k \in L(G - j_k P)$ we have $j_k \leq \deg G$.

Example:

- We consider the Hermitian curve over $\mathbb{F}_4$ defined by $x_2^2 + x_2 = x_1^3$.
- Write $\mathbb{F}_4 = \mathbb{F}_2[\alpha]$ with $\alpha^2 = \alpha + 1$.

Example

- Write $P_1 = (0,0)$, $P_2 = (0,1)$, $P_3 = (1,\alpha)$, $P_4 = (1,\alpha^2)$, $P_5 = (\alpha,\alpha)$, $P_6 = (\alpha,\alpha^2)$, $P_7 = (\alpha^2,\alpha)$,$P_8 = (\alpha^2,\alpha^2)$ and denote by T_∞ the unique pole of x_1.
- We now take $D = P_1 + \cdots + P_8$, $G = 4T_\infty$, and $A = 35T_\infty$.

Example

- Write $P_1 = (0,0)$, $P_2 = (0,1)$, $P_3 = (1,\alpha)$, $P_4 = (1,\alpha^2)$, $P_5 = (\alpha,\alpha)$, $P_6 = (\alpha,\alpha^2)$, $P_7 = (\alpha^2,\alpha)$,$P_8 = (\alpha^2,\alpha^2)$ and denote by T_∞ the unique pole of x_1.
- We now take $D = P_1 + \cdots + P_8$, $G = 4T_\infty$, and $A = 35T_\infty$.
- If we choose $s = 6$ and $\lambda = 8$, we can correct 2 errors using the list decoder.

Example

- Write $P_1 = (0,0)$, $P_2 = (0,1)$, $P_3 = (1,\alpha)$, $P_4 = (1,\alpha^2)$, $P_5 = (\alpha,\alpha)$, $P_6 = (\alpha,\alpha^2)$, $P_7 = (\alpha^2,\alpha)$,$P_8 = (\alpha^2,\alpha^2)$ and denote by T_∞ the unique pole of x_1.
- We now take $D = P_1 + \cdots + P_8$, $G = 4T_\infty$, and $A = 35T_\infty$.
- If we choose $s = 6$ and $\lambda = 8$, we can correct 2 errors using the list decoder.
- In order to describe the list-decoding procedure, we need to choose bases for the spaces $L(A - iG)$, whose dimension we denote by l_i.

Example

- In this case we can for $0 \leq i \leq \lambda$ and $1 \leq j \leq l_i$ choose

$$g_{ij} = \begin{cases} 1 & \text{if } j = 1, \\ x_1 x_2^{(j-2)/3} & \text{if } j \equiv 2 \bmod 3, \\ x_2^{j/3} & \text{if } j \equiv 0 \bmod 3, \\ x_1^2 x_2^{(j-4)/3} & \text{if } j > 1 \text{ and } j \equiv 1 \bmod 3. \end{cases}$$

Example

- In this case we can for $0 \leq i \leq \lambda$ and $1 \leq j \leq l_i$ choose

$$g_{ij} = \begin{cases} 1 & \text{if } j = 1, \\ x_1 x_2^{(j-2)/3} & \text{if } j \equiv 2 \bmod 3, \\ x_2^{j/3} & \text{if } j \equiv 0 \bmod 3, \\ x_1^2 x_2^{(j-4)/3} & \text{if } j > 1 \text{ and } j \equiv 1 \bmod 3. \end{cases}$$

- Suppose that we transmit the all zero word and receive.

$$(\alpha^2, 0, 0, \alpha^2, 0, 0, 0, 0).$$

- The majority voting decoder fails to decode this word, but we can use list decoding if we choose $s = 6$ and $\lambda = 8$.

Example: The interpolation polynomial

One can find (e.g. using linear algebra, 168 equations and 171 variables) an interpolation polynomial:

$$Q(y) = (1 + x_2 + \alpha x_2^2 + \alpha x_1^2 x_2 + \alpha^2 x_1 x_2^2 + \alpha x_2^3 + \alpha^2 x_1^2 x_2^2 + \alpha x_1 x_2^3 + x_2^4 + \alpha x_1^2 x_2^3 + \alpha^2 x_1 x_2^4 + x_1^2 x_2^4 + \alpha x_1 x_2^5 + \alpha^2 x_1^2 x_2^5 + \alpha x_1 x_2^6 + x_2^7 + \alpha x_1^2 x_2^6 + x_1 x_2^7 + x_2^8 + x_1^2 x_2^7 + \alpha x_1 x_2^8 + \alpha x_2^9 + \alpha^2 x_1^2 x_2^8 + x_1 x_2^9 + \alpha^2 x_2^{10} + x_1^2 x_2^9) y + (\alpha^2 + \alpha x_1 + \alpha x_1^2 + x_2^2 + \alpha^2 x_1^2 x_2 + \alpha^2 x_2^3 + x_1^2 x_2^2 + \alpha^2 x_1 x_2^3 + \alpha^2 x_2^5 + x_1^2 x_2^4 + x_1^2 x_2^4 + \alpha^2 x_2^6 + \alpha x_1^2 x_2^5 + \alpha x_2^7 + \alpha^2 x_1^2 x_2^6 + \alpha x_1 x_2^7 + x_2^8 + \alpha^2 x_1 x_2^8 + \alpha x_2^9) y^2 + (\alpha^2 + \alpha x_2 + x_1 x_2 + \alpha^2 x_1^2 x_2 + x_1 x_2^2 + \alpha x_2^3 + x_1^2 x_2^2 + \alpha^2 x_2^4 + \alpha^2 x_1^2 x_2^3 + \alpha x_2^5 + \alpha x_1^2 x_2^4 + \alpha^2 x_1 x_2^5 + \alpha x_2^6 + \alpha^2 x_1^2 x_2^5 + \alpha^2 x_1 x_2^6) y^3 + (\alpha + x_1 + \alpha^2 x_2 + x_1 x_2 + \alpha x_2^2 + \alpha^2 x_1^2 x_2 + \alpha x_1 x_2^2 + x_2^3 + \alpha x_1 x_2^3 + \alpha x_2^4 + x_1^2 x_2^3) y^4 + (\alpha + \alpha^2 x_2 + \alpha^2 x_1 x_2 + x_2^2 + x_1^2 x_2 + x_1 x_2^2 + \alpha^2 x_1^2 x_2^2 + \alpha x_1 x_2^3) y^5 + (1 + \alpha^2 x_1 + \alpha x_2 + \alpha^2 x_1^2 + \alpha^2 x_1 x_2 + x_2^2 + \alpha^2 x_1^2 x_2) y^6 + y^7 + (\alpha^2 + \alpha x_1) y^8.$$

Example: Finding factors in $Q(y)$

- In order to factorize this using the first method described above, we let

$$\mathbb{F}_{4^3} = \mathbb{F}_4[X_2]/\langle {X_2}^3+\alpha X_2+1\rangle, \quad \mathbb{F}_{4^{3\times 3}} = \mathbb{F}_{4^3}[X_1]/\langle {X_1}^3+{X_2}^2+X_2\rangle.$$

- This makes sense since the polynomial ${X_2}^3 + \alpha X_2 + 1$ is irreducible over $\mathbb{F}_4$ and for any root X_2 of it, the polynomial ${X_1}^3 + {X_2}^2 + X_2$ is irreducible over $\mathbb{F}_{4^3}$.

Example: Finding factors in $Q(y)$

- In order to factorize this using the first method described above, we let

$$\mathbb{F}_{4^3} = \mathbb{F}_4[X_2]/\langle {X_2}^3 + \alpha X_2 + 1\rangle, \quad \mathbb{F}_{4^{3\times 3}} = \mathbb{F}_{4^3}[X_1]/\langle {X_1}^3 + {X_2}^2 + X_2\rangle.$$

- This makes sense since the polynomial ${X_2}^3 + \alpha X_2 + 1$ is irreducible over $\mathbb{F}_4$ and for any root X_2 of it, the polynomial ${X_1}^3 + {X_2}^2 + X_2$ is irreducible over $\mathbb{F}_{4^3}$.
- If we let R be a point (x_1, x_2) on the curve in $\mathbb{F}_{4^{3\times 3}}$ corresponding to the description above we get:

Example: Finding factors in $Q(y)$

$$
\begin{aligned}
&\widehat{Q}(y) = \\
&(\,(\alpha + \alpha x_2) + (\alpha x_2 + \alpha^2 x_2^2)\,x_1 + (\alpha x_2 + x_2^2)\,x_1^2\,)\,y + (\,(\alpha + \alpha^2 x_2) + \\
&(\alpha+\alpha^2 x_2)x_1+(1+\alpha x_2^2)x_1^2)y^2+((\alpha^2 x_2+\alpha^2 x_2^2)+(\alpha+\alpha x_2+\alpha^2 x_2^2)x_1+ \\
&(\alpha^2 + \alpha x_2 + \alpha^2 x_2^2)x_1^2)y^3+((\alpha^2 + x_2 + \alpha x_2^2)+(\alpha^2 + \alpha^2 x_2 + \alpha x_2^2)x_1+ \\
&(\alpha x_2 + x_2^2)\,x_1^2\,)\,y^4 + (\,(\alpha + x_2) + (\alpha + \alpha\, x_2^2)\,x_1 + (1 + \alpha\, x_2)\,x_1^2\,)\,y^5 + \\
&(\,(x_2 + \alpha x_2^2) + (1 + \alpha x_2 + x_2^2)x_1 + (\alpha + \alpha^2 x_2^2)x_2^2\,)y^6 + ((1 + \alpha^2 x_2^2)+ \\
&(\alpha^2 + \alpha x_2^2)x_1 + (\alpha + x_2^2)x_1^2)y^7 + y^8.
\end{aligned}
$$

Example: Finding factors in $Q(y)$

$\widehat{Q}(y) =$
$((\alpha+\alpha x_2)+(\alpha x_2+\alpha^2 x_2^2)x_1+(\alpha x_2+x_2^2)x_1^2)y+((\alpha+\alpha^2 x_2)+(\alpha+\alpha^2 x_2)x_1+(1+\alpha x_2^2)x_1^2)y^2+((\alpha^2 x_2+\alpha^2 x_2^2)+(\alpha+\alpha x_2+\alpha^2 x_2^2)x_1+(\alpha^2+\alpha x_2+\alpha^2 x_2^2)x_1^2)y^3+((\alpha^2+x_2+\alpha x_2^2)+(\alpha^2+\alpha^2 x_2+\alpha x_2^2)x_1+(\alpha x_2+x_2^2)x_1^2)y^4+((\alpha+x_2)+(\alpha+\alpha x_2^2)x_1+(1+\alpha x_2)x_1^2)y^5+((x_2+\alpha x_2^2)+(1+\alpha x_2+x_2^2)x_1+(\alpha+\alpha^2 x_2^2)x_2^2)y^6+((1+\alpha^2 x_2^2)+(\alpha^2+\alpha x_2^2)x_1+(\alpha+x_2^2)x_1^2)y^7+y^8.$

This polynomial has three factors of degree one:

$$y$$
$$(\alpha^2+\alpha^2 x_1+\alpha^2 x_1^2)+y$$
$$((\alpha^2+\alpha x_2+x_2^2)+(\alpha x_2+\alpha^2 x_2^2)x_1+(1+\alpha^2 x_2+\alpha x_2^2)x_1^2)+y$$

Example: Finding factors in $Q(y)$

- The last of these factors does not correspond to a codeword since it is not in $L(G)$ but the first two factors correspond to the codewords

$$(\alpha^2, \alpha^2, \alpha^2, \alpha^2, 0, 0, 0, 0)$$
$$(0, 0, 0, 0, 0, 0, 0, 0)$$

which both have distance two to the received word.

Example: Finding factors in $Q(y)$

- The last of these factors does not correspond to a codeword since it is not in $L(G)$ but the first two factors correspond to the codewords

$$(\alpha^2, \alpha^2, \alpha^2, \alpha^2, 0, 0, 0, 0)$$
$$(0, 0, 0, 0, 0, 0, 0, 0)$$

which both have distance two to the received word.

- Now we shall describe the Hensel-lifting approach to find y-roots of $Q(y)$.
- As the point in which we expand, we choose $P = P_{00}$ and as local parameter for P we pick $t = x_1$.
- Then we write $Q(y)$ explicitly as an element of $\mathbb{F}_4[[t]][y]$.

Example: Finding factors in $Q(y)$

- Since $x_1 = t$, we find from the defining equation of the curve that $x_2 = t^3 + t^6 + t^{12} + \mathcal{O}(t^{24})$.

Example: Finding factors in $Q(y)$

- Since $x_1 = t$, we find from the defining equation of the curve that $x_2 = t^3 + t^6 + t^{12} + \mathcal{O}(t^{24})$.
- Substituting this in $Q(y)$ we see that

$$
\begin{aligned}
&Q(y) = \\
&(1 + t^3 + \alpha t^5 + \alpha^2 t^6 + \alpha^2 t^7 + t^8 + \alpha t^9)y + \\
&(\alpha^2 + \alpha t + \alpha t^2 + \alpha^2 t^5 + t^6 + \alpha t^8 + \alpha^2 t^9)y^2 + \\
&(\alpha^2 + \alpha t^3 + t^4 + \alpha^2 t^5 + \alpha t^6 + \alpha t^8 + \alpha t^9)y^3 + \\
&(\alpha + t + \alpha^2 t^3 + t^4 + \alpha^2 t^5 + t^6 + \alpha^2 t^7 + \alpha^2 t^8 + t^9)y^4 + \\
&(\alpha + \alpha^2 t^3 + \alpha^2 t^4 + t^5 + \alpha t^6 + \alpha t^7 + \alpha t^8)y^5 + \\
&(1 + \alpha^2 t + \alpha^2 t^2 + \alpha t^3 + \alpha^2 t^4 + \alpha^2 t^5 + \alpha^2 t^6 + \alpha^2 t^7 + \alpha^2 t^8)y^6 + \\
&y^7 + (\alpha^2 + \alpha t)y^8 + \mathcal{O}(t^{10}).
\end{aligned}
$$

Example: Finding factors in $Q(y)$

- We can now find all possible values of a_0, as roots of $Q_0(0, y) = \alpha^2 y(y - \alpha)(y - \alpha^2)^6$.
- Therefore there are three possibilities for a_0: 0, α and α^2.

Example: Finding factors in $Q(y)$

- We can now find all possible values of a_0, as roots of $Q_0(0, y) = \alpha^2 y(y - \alpha)(y - \alpha^2)^6$.
- Therefore there are three possibilities for a_0: 0, α and α^2.
- For each of them separately we can calculate the updated polynomial $Q_1(t, y)$.
- If a_0 equals 0 or α, it has multiplicity 1, implying by Lemma 22 that the next coefficient is the root of a polynomial of degree at most one, i.e. a_1 is uniquely determined if it exists.
- Since $a_0 = \alpha^2$ has multiplicity 6 this need not be true in that case.

Example: Finding factors in $Q(y)$

- For $a_0 = \alpha^2$ we get $Q_1(t, y) = t^{-6} Q_0(t, ty + \alpha^2)$ and

$$Q_1(t, y) = \\ 1 + t^3 + (t + \alpha t^2 + \alpha^2 t^3)y + (1 + \alpha^2 t + \alpha t^2 + \alpha t^3)y^2 + \\ (\alpha + t + \alpha^2 t^2 + \alpha t^3)y^3 + (1 + \alpha t + \alpha t^2 + t^3)y^4 + (\alpha^2 t^2 + \alpha^2 t^3)y^5 + \\ (\alpha + \alpha^2 t + \alpha^2 t^2 + \alpha t^3)y^6 + ty^7 + (\alpha^2 t^2 + \alpha t^3)y^8 + \mathcal{O}(t^4)$$

Example: Finding factors in $Q(y)$

- For $a_0 = \alpha^2$ we get $Q_1(t,y) = t^{-6}Q_0(t, ty+\alpha^2)$ and

$$Q_1(t,y) = \\ 1+t^3+(t+\alpha t^2+\alpha^2 t^3)y+(1+\alpha^2 t+\alpha t^2+\alpha t^3)y^2+ \\ (\alpha+t+\alpha^2 t^2+\alpha t^3)y^3+(1+\alpha t+\alpha t^2+t^3)y^4+(\alpha^2 t^2+\alpha^2 t^3)y^5+ \\ (\alpha+\alpha^2 t+\alpha^2 t^2+\alpha t^3)y^6+ty^7+(\alpha^2 t^2+\alpha t^3)y^8+\mathcal{O}(t^4)$$

- This gives

$$Q_1(0,y) = (y-\alpha)(y-\alpha^2)(\alpha y^4+\alpha y^3+y^2+y+1).$$

- We see that if $a_0 = \alpha^2$, then $a_1 = \alpha$ or $a_1 = \alpha^2$ both having multiplicity one. The degree 4 factor of $Q_1(0,y)$ does not give $\mathbb{F}_4$-rational solutions and is therefore discarded.

Example: Finding factors in $Q(y)$

- For $a_0 = \alpha^2$ we get $Q_1(t,y) = t^{-6}Q_0(t, ty+\alpha^2)$ and

$$Q_1(t,y) = \\ 1 + t^3 + (t + \alpha t^2 + \alpha^2 t^3)y + (1 + \alpha^2 t + \alpha t^2 + \alpha t^3)y^2 + \\ (\alpha + t + \alpha^2 t^2 + \alpha t^3)y^3 + (1 + \alpha t + \alpha t^2 + t^3)y^4 + (\alpha^2 t^2 + \alpha^2 t^3)y^5 + \\ (\alpha + \alpha^2 t + \alpha^2 t^2 + \alpha t^3)y^6 + ty^7 + (\alpha^2 t^2 + \alpha t^3)y^8 + \mathcal{O}(t^4)$$

- This gives

$$Q_1(0,y) = (y-\alpha)(y-\alpha^2)(\alpha y^4 + \alpha y^3 + y^2 + y + 1).$$

- We see that if $a_0 = \alpha^2$, then $a_1 = \alpha$ or $a_1 = \alpha^2$ both having multiplicity one. The degree 4 factor of $Q_1(0,y)$ does not give $\mathbb{F}_4$-rational solutions and is therefore discarded.
- The outcome of the entire Hensel-lifting procedure including multiplicities and values of the a_i's can be described in a tree structure.

Example: Tree structure of Hensel lifting

Example: Tree structure of Hensel lifting

- Thus we get four outputs for (a_0, a_1, a_2, a_3) in all:

$$\begin{gathered}(\alpha^2, \alpha^2, \alpha^2, 0),\\ (\alpha^2, \alpha, \alpha^2, 1),\\ (\alpha, 1, 1, \alpha^2),\\ (0, 0, 0, 0).\end{gathered}$$

Example: Tree structure of Hensel lifting

- Thus we get four outputs for (a_0, a_1, a_2, a_3) in all:

$$(\alpha^2, \alpha^2, \alpha^2, 0),$$
$$(\alpha^2, \alpha, \alpha^2, 1),$$
$$(\alpha, 1, 1, \alpha^2),$$
$$(0, 0, 0, 0).$$

- The corresponding functions are

$$\alpha^2 + \alpha^2 x + \alpha^2 x^2,$$
$$\alpha^2 + \alpha x + \alpha^2 x^2 + y,$$
$$\alpha + x + x^2 + \alpha^2,$$
$$0.$$

Example: Tree structure of Hensel lifting

- Thus we get four outputs for (a_0, a_1, a_2, a_3) in all:
$$(\alpha^2, \alpha^2, \alpha^2, 0),$$
$$(\alpha^2, \alpha, \alpha^2, 1),$$
$$(\alpha, 1, 1, \alpha^2),$$
$$(0, 0, 0, 0).$$
- The corresponding functions are
$$\alpha^2 + \alpha^2 x + \alpha^2 x^2,$$
$$\alpha^2 + \alpha x + \alpha^2 x^2 + y,$$
$$\alpha + x + x^2 + \alpha^2,$$
$$0.$$
- The first and the last function give rise to solutions of the equation $Q(f) = 0$ and thus to two codewords, while the remaining two are not solutions.

Contents

Syndrome formulation of list decoding

- The list decoding algorithm can be reformulated in terms of syndromes.
- As for the basic algorithm, the advantage is that variables are eliminated from the system of linear equations used to determine the interpolation polynomial.
- As before, we are interested in finding an interpolation polynomial $Q(y) = \sum_{i=0}^{\lambda} Q_i y^i$ such that $Q_i \in L(A - iG)$ and such that (P_l, r_l) is a zero of $Q(y)$ of multiplicity s for all i between 1 and n.

Syndrome formulation of list decoding

- Let $g_{i1}, \ldots, g_{il_i}$ be a basis of $L(A - iG)$ and write $Q_i = \sum_{j=1}^{l_i} q_{ij} g_{ij}$.
- The condition that (P_l, r_l) is a zero of $Q(y)$ of multiplicity s gives rise to $\binom{s+1}{2}$ linear equations in the coefficients q_{ij}.

Syndrome formulation of list decoding

- Let $g_{i1}, \dots, g_{il_i}$ be a basis of $L(A - iG)$ and write $Q_i = \sum_{j=1}^{l_i} q_{ij} g_{ij}$.
- The condition that (P_l, r_l) is a zero of $Q(y)$ of multiplicity s gives rise to $\binom{s+1}{2}$ linear equations in the coefficients q_{ij}.
- More explicitly: first for any $P_l \in \operatorname{supp} D$ choose a function $t_l \in \mathscr{F}$ such that $v_{P_l}(t_l) = 1$. Given such a t_l, we can write a function g that is regular at P_l as a power series in t_l, say

$$g = \alpha_0 + \alpha_1 t + \cdots + \alpha_a t^a + \cdots .$$

Syndrome formulation of list decoding

- Let $g_{i1}, \ldots, g_{il_i}$ be a basis of $L(A - iG)$ and write $Q_i = \sum_{j=1}^{l_i} q_{ij} g_{ij}$.
- The condition that (P_l, r_l) is a zero of $Q(y)$ of multiplicity s gives rise to $\binom{s+1}{2}$ linear equations in the coefficients q_{ij}.
- More explicitly: first for any $P_l \in \operatorname{supp} D$ choose a function $t_l \in \mathscr{F}$ such that $v_{P_l}(t_l) = 1$. Given such a t_l, we can write a function g that is regular at P_l as a power series in t_l, say

$$g = \alpha_0 + \alpha_1 t + \cdots + \alpha_a t^a + \cdots .$$

- We have $\alpha_0 = g(P_l)$. The α_a depend in general on P_l and the choice of $t_l \in \mathscr{F}$.

Syndrome formulation of list decoding

- Let $g_{i1}, \ldots, g_{il_i}$ be a basis of $L(A - iG)$ and write $Q_i = \sum_{j=1}^{l_i} q_{ij} g_{ij}$.
- The condition that (P_l, r_l) is a zero of $Q(y)$ of multiplicity s gives rise to $\binom{s+1}{2}$ linear equations in the coefficients q_{ij}.
- More explicitly: first for any $P_l \in \operatorname{supp} D$ choose a function $t_l \in \mathscr{F}$ such that $v_{P_l}(t_l) = 1$. Given such a t_l, we can write a function g that is regular at P_l as a power series in t_l, say

$$g = \alpha_0 + \alpha_1 t + \cdots + \alpha_a t^a + \cdots .$$

- We have $\alpha_0 = g(P_l)$. The α_a depend in general on P_l and the choice of $t_l \in \mathscr{F}$.
- Let $D_{t_l}^{(a)}$ be the a-th Hasse-derivative with respect to t_l, then $D_{t_l}^{(a)}(g)(P) = \alpha_a$.

Hasse–derivative

- We extend the Hasse-derivative to $\mathscr{F}[y]$ by

$$D_y^{(b)} D_{t_l}^{(a)}(gy^j) := \binom{j}{b} D_{t_l}^{(a)}(g) y^{j-b},$$

and extending it linearly to all of $\mathscr{F}[y]$.

- If we expand the polynomial $Q(y)$ as a power series in the variables t_l and $y - r_l$, then with this definition the coefficient of $t_l^a (y - r_l)^b$ is given exactly by $D_y^{(b)} D_{t_l}^{(a)}(Q(y))(P_l, r_l)$.

Hasse–derivative

- We extend the Hasse-derivative to $\mathscr{F}[y]$ by

$$D_y^{(b)} D_{t_l}^{(a)}(gy^j) := \binom{j}{b} D_{t_l}^{(a)}(g) y^{j-b},$$

  and extending it linearly to all of $\mathscr{F}[y]$.
- If we expand the polynomial $Q(y)$ as a power series in the variables t_l and $y - r_l$, then with this definition the coefficient of $t_l^a (y - r_l)^b$ is given exactly by $D_y^{(b)} D_{t_l}^{(a)}(Q(y))(P_l, r_l)$.
- By the approximation theorem there exists $t \in \mathscr{F}$ such that $v_P(t) = 1$ for all $P \in \operatorname{supp} D$. Thus from now on we assume that $t_l = t$ does not depend on l.

Hasse–derivative

- We extend the Hasse-derivative to $\mathscr{F}[y]$ by

$$D_y^{(b)} D_{t_l}^{(a)}(gy^j) := \binom{j}{b} D_{t_l}^{(a)}(g) y^{j-b},$$

  and extending it linearly to all of $\mathscr{F}[y]$.
- If we expand the polynomial $Q(y)$ as a power series in the variables t_l and $y - r_l$, then with this definition the coefficient of $t_l^a (y - r_l)^b$ is given exactly by $D_y^{(b)} D_{t_l}^{(a)}(Q(y))(P_l, r_l)$.
- By the approximation theorem there exists $t \in \mathscr{F}$ such that $v_P(t) = 1$ for all $P \in \operatorname{supp} D$. Thus from now on we assume that $t_l = t$ does not depend on l.
- The equations saying that (P_l, r_l) should be a zero of multiplicity s in $Q(y)$ are then:

$$D_y^{(b)} D_t^{(a)}(Q(y))(P_l, r_l) = 0, \text{ for all } a, b \text{ with } a + b < s.$$

Reformulating the linear system

- The interpolation conditions are thus equivalent to:

$$\sum_{i=b}^{\lambda} \binom{i}{b} r_l^{i-b} \sum_{j=1}^{l_i} q_{ij} D_t^{(a)}(g_{ij})(P_l) = 0, \qquad (24)$$

for all $\binom{s+1}{2}$ pairs of nonnegative integers (a, b) such that $a + b < s$.

Reformulating the linear system

- The interpolation conditions are thus equivalent to:

$$\sum_{i=b}^{\lambda} \binom{i}{b} r_l^{i-b} \sum_{j=1}^{l_i} q_{ij} D_t^{(a)}(g_{ij})(P_l) = 0, \qquad (24)$$

for all $\binom{s+1}{2}$ pairs of nonnegative integers (a, b) such that $a + b < s$.

- As before, we would like to write these equations in matrix form

$$\mathbf{M} \begin{pmatrix} \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \qquad (25)$$

Matrices

For $0 \leq b \leq s-1$ and $b \leq i \leq \lambda$, we therefore introduce the following $(s-b)n \times l_i$ matrix:

$$\mathbf{M}_i^{(i-b)} := \begin{pmatrix} g_{i1}(P_1) & \cdots & g_{il_i}(P_1) \\ \vdots & & \vdots \\ D_t^{(s-1-b)}(g_{i1})(P_1) & \cdots & D_t^{(s-1-b)}(g_{il_i})(P_1) \\ & & \\ \vdots & & \vdots \\ & & \\ g_{i1}(P_n) & \cdots & g_{il_i}(P_n) \\ \vdots & & \vdots \\ D_t^{(s-1-b)}(g_{i1})(P_n) & \cdots & D_t^{(s-1-b)}(g_{il_i})(P_n) \end{pmatrix}$$

Matrices

We also introduce the $(s-b)n \times (s-b)n$ matrix:

$$\mathbf{D}_i^{(b)} := \binom{i+b}{b} \begin{pmatrix} r_1^i & & & & & & \\ & \ddots & & & & & \\ & & r_1^i & & & & \\ & & & \ddots & & & \\ & & & & r_n^i & & \\ & & & & & \ddots & \\ & & & & & & r_n^i \end{pmatrix},$$

where every element r_l^i is repeated $s-b$ times on the diagonal.

Matrices

Using these, we can then find the desired matrix $\mathbf{M}$:

$$\left[\begin{array}{c|c|c|c|c|c}
\mathbf{M}_0^{(0)} & \mathbf{D}_1^{(0)}\mathbf{M}_1^{(1)} & \cdots & \mathbf{D}_{s-1}^{(0)}\mathbf{M}_{s-1}^{(s-1)} & \cdots & \mathbf{D}_{\lambda}^{(0)}\mathbf{M}_{\lambda}^{(\lambda)} \\ \hline
\mathbf{0} & \mathbf{M}_1^{(0)} & \cdots & \mathbf{D}_{s-2}^{(1)}\mathbf{M}_{s-1}^{(s-2)} & \cdots & \mathbf{D}_{\lambda-1}^{(1)}\mathbf{M}_{\lambda}^{(\lambda-1)} \\ \hline
\vdots & \ddots & \ddots & \vdots & & \vdots \\ \hline
\mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}_{s-1}^{(0)} & \cdots & \mathbf{D}_{\lambda-s+1}^{(s-1)}\mathbf{M}_{\lambda}^{(\lambda-s+1)}
\end{array}\right].$$

With this $\mathbf{M}$, we can reformulate the interpolation equations as matrix equation (25).

Matrices

Using these, we can then find the desired matrix $\mathbf{M}$:

$$\left[\begin{array}{c|c|c|c|c|c}
\mathbf{M}_0^{(0)} & \mathbf{D}_1^{(0)}\mathbf{M}_1^{(1)} & \cdots & \mathbf{D}_{s-1}^{(0)}\mathbf{M}_{s-1}^{(s-1)} & \cdots & \mathbf{D}_\lambda^{(0)}\mathbf{M}_\lambda^{(\lambda)} \\ \hline
\mathbf{0} & \mathbf{M}_1^{(0)} & \cdots & \mathbf{D}_{s-2}^{(1)}\mathbf{M}_{s-1}^{(s-2)} & \cdots & \mathbf{D}_{\lambda-1}^{(1)}\mathbf{M}_\lambda^{(\lambda-1)} \\ \hline
\vdots & \ddots & \ddots & \vdots & & \vdots \\ \hline
\mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}_{s-1}^{(0)} & \cdots & \mathbf{D}_{\lambda-s+1}^{(s-1)}\mathbf{M}_\lambda^{(\lambda-s+1)}
\end{array}\right].$$

With this $\mathbf{M}$, we can reformulate the interpolation equations as matrix equation (25).

Example:
We show how to calculate the above equations in case of the Hermitian curve given by the equation $x_2^q + x_2 = x_1^{q+1}$ defined over $\mathbb{F}_{q^2}$.

Example: The Hermitian Curve

- $t = x^{q^2} - x$ is a local parameter for all points on the curve other than T_∞.
- We wish to compute $D_t^{(a)}(f)$ for any function $f \in \mathscr{F}$.
- Hasse derivatives satisfy the Leibniz rule:

$$D_t^{(a)}(f_1 \cdots f_m) = \sum_{i_1+\cdots+i_m=a} D_t^{(i_1)}(f_1) \cdots D_t^{(i_m)}(f_m).$$

Example: The Hermitian Curve

- $t = x^{q^2} - x$ is a local parameter for all points on the curve other than T_∞.
- We wish to compute $D_t^{(a)}(f)$ for any function $f \in \mathscr{F}$.
- Hasse derivatives satisfy the Leibniz rule:

$$D_t^{(a)}(f_1 \cdots f_m) = \sum_{i_1 + \cdots + i_m = a} D_t^{(i_1)}(f_1) \cdots D_t^{(i_m)}(f_m).$$

- Using this and the linearity of Hasse derivatives, we see that it is enough to compute $D_t^{(a)}(x_1)$ and $D_t^{(a)}(x_2)$ for all natural numbers a.
- We will now show how to calculate $D_t^{(a)}(x_1)$ recursively. We have that $D_t^{(0)}(x_1) = x_1$. Now suppose that $a > 0$ and that we know $D_t^{(\alpha)}(x_1)$ for all α between 0 and $a - 1$.

Example: The Hermitian Curve

- Using the equation $t = x_1^{q^2} + x_1$, it follows that $D_t^{(a)}(x_1) = D_t^{(a)}(t) - D_t^{(a)}(x_1^{q^2})$.
- $D_t^{(0)}(t) = t$, $D_t^{(1)}(t) = 1$ and $D_t^{(a)}(t) = 0$ if $a > 1$.
- By Leibniz rule:

$$D_t^{(a)}(x_1^{q^2}) = \sum_{i_1 + \cdots i_{q^2} = a} D_t^{i_1}(x_1) \cdots D_t^{(i_{q^2})}(x_1).$$

If $i_j = a$ for some j, the remaining indices are zero implying that for this choice of indices we find the term $x_1^{a-1} D_t^{(a)}(x_1)$.

Example: The Hermitian Curve

- Using the equation $t = x_1^{q^2} + x_1$, it follows that $D_t^{(a)}(x_1) = D_t^{(a)}(t) - D_t^{(a)}(x_1^{q^2})$.
- $D_t^{(0)}(t) = t$, $D_t^{(1)}(t) = 1$ and $D_t^{(a)}(t) = 0$ if $a > 1$.
- By Leibniz rule:

$$D_t^{(a)}(x_1^{q^2}) = \sum_{i_1 + \cdots i_{q^2} = a} D_t^{i_1}(x_1) \cdots D_t^{(i_{q^2})}(x_1).$$

  If $i_j = a$ for some j, the remaining indices are zero implying that for this choice of indices we find the term $x_1^{a-1} D_t^{(a)}(x_1)$.
- By varying j between 1 and q^2, we see that there are exactly q^2 such terms. Thus these terms do not contribute to the sum.
- This means that $D_t^{(a)}(x_1) = D_t^{(a)}(t - x_1^{q^2})$ can be expressed as polynomial in $D_t^{(\alpha)}(x_1)$ for α varying between 0 and $a - 1$.

Example: The Hermitian Curve

- It remains to show how to calculate $D_t^{(a)}(x_2)$ recursively. First $D_t^{(0)}(x_2) = x_2$ and since $x_2^q + x_2 = x_1^{q+1}$, we also have that $D_t^{(a)}(x_2) = D_t^{(a)}(x_1^{q+1}) - D_t^{(a)}(x_2^q)$.

Example: The Hermitian Curve

- It remains to show how to calculate $D_t^{(a)}(x_2)$ recursively. First $D_t^{(0)}(x_2) = x_2$ and since $x_2^q + x_2 = x_1^{q+1}$, we also have that $D_t^{(a)}(x_2) = D_t^{(a)}(x_1^{q+1}) - D_t^{(a)}(x_2^q)$.
- We already know how to calculate $D_t^{(a)}(x_1^{q+1})$ recursively and as before we can express $D_t^{(a)}(x_2^q)$ as a polynomial in $D_t^{(\alpha)}(x_2)$ with α between 0 and $a-1$.

Example: The Hermitian Curve

- It remains to show how to calculate $D_t^{(a)}(x_2)$ recursively. First $D_t^{(0)}(x_2) = x_2$ and since $x_2^q + x_2 = x_1^{q+1}$, we also have that $D_t^{(a)}(x_2) = D_t^{(a)}(x_1^{q+1}) - D_t^{(a)}(x_2^q)$.
- We already know how to calculate $D_t^{(a)}(x_1^{q+1})$ recursively and as before we can express $D_t^{(a)}(x_2^q)$ as a polynomial in $D_t^{(\alpha)}(x_2)$ with α between 0 and $a-1$.
- For future use, we state some explicit results for $q = 2$:

a	0	1	2	3	4	5
$D_t^{(a)}(x_1)$	x_1	1	0	0	1	0
$D_t^{(a)}(x_2)$	x_2	x_1^2	$x_1 + x_1^4$	1	x_1^8	0

Interpreatation as generator matrices

We now establish some facts on the matrices $\mathbf{M}_i^{(0)}$. We will think about them as generator matrices of certain codes:

Interpreatation as generator matrices

We now establish some facts on the matrices $\mathbf{M}_i^{(0)}$. We will think about them as generator matrices of certain codes:

Definition

Let s and $D = P_1 + \cdots + P_n$ be as before. Let A be a divisor of arbitrary degree with $\operatorname{supp} A \cap \operatorname{supp} D = \emptyset$. Further, let $t \in \mathscr{F}$ be a local parameter for all $P \in \operatorname{supp} D$. We define

$$\begin{array}{rccl} \mathrm{Ev}_P^{(s)}: & L(A) & \to & \mathbb{F}^s \\ & f & \mapsto & (f(P), D_t^{(1)}(f)(P), \ldots, D_t^{(s-1)}(f)(P)) \\ \mathrm{Ev}_D^{(s)}: & L(A) & \to & \mathbb{F}^{sn} \\ & f & \mapsto & (\mathrm{Ev}_{P_1}^{(s)}(f), \ldots, \mathrm{Ev}_{P_n}^{(s)}(f)) \end{array}$$

and $C_L^{(s)}(D, A) := \mathrm{Ev}_D^{(s)}(L(A))$.

Interpreatation as generator matrices

- Note that if $s > 1$, the map $\mathrm{Ev}_P^{(s)}$ depends on the choice of the local parameter t.
- The point of the definition is that the columns occurring in the matrix $\mathbf{M}_i^{(0)}$ are codewords in the code $C_L^{(s-i)}(D, A - iG)$.
- Also: $\mathrm{rank}\,\mathbf{M}_i^{(0)} = \dim C_L^{(s-i)}(A - iG)$.

Interpreatation as generator matrices

- Note that if $s > 1$, the map $\mathrm{Ev}_P^{(s)}$ depends on the choice of the local parameter t.
- The point of the definition is that the columns occurring in the matrix $\mathbf{M}_i^{(0)}$ are codewords in the code $C_L^{(s-i)}(D, A - iG)$.
- Also: $\mathrm{rank}\,\mathbf{M}_i^{(0)} = \dim C_L^{(s-i)}(A - iG)$.
- In order to define the analogue of the code $C_\Omega(D, A)$, we consider a differential $\omega \in \Omega(-sD + A)$. Locally at a point $P \in \mathrm{supp}\, D$, one can then write

$$\omega = (\beta_s t^{-s} + \cdots + \beta_1 t^{-1} + \cdots)\, dt.$$

- We can calculate β_i using residues, as $\beta_i = \mathrm{res}_P(t^{i-1}\omega)$. This motivates the following definition:

Dual codes

Definition

Let s, D, A and t be as in Definition 24. We define

$$\begin{array}{rrcl} \mathrm{Res}_P^{(s)}: & \Omega(-sD+A) & \rightarrow & \mathbb{F}^s \\ & \omega & \mapsto & (\mathrm{res}_P(\omega), \mathrm{res}_P(t\omega), \ldots, \mathrm{res}_P(t^{s-1}\omega)), \\ \mathrm{Res}_D^{(s)}: & \Omega(-sD+A) & \rightarrow & \mathbb{F}^{sn} \\ & \omega & \mapsto & (\mathrm{Res}_{P_1}^{(s)}(\omega), \ldots, \mathrm{Res}_{P_n}^{(s)}(\omega)) \end{array}$$

and $C_\Omega^{(s)}(D, A) := \mathrm{Res}_D^{(s)}(\Omega(-sD+A))$.

Dual codes

Definition

Let s, D, A and t be as in Definition 24. We define

$$
\begin{array}{rcll}
\mathrm{Res}_P^{(s)} : & \Omega(-sD+A) & \rightarrow & \mathbb{F}^s \\
& \omega & \mapsto & (\mathrm{res}_P(\omega), \mathrm{res}_P(t\omega), \ldots, \mathrm{res}_P(t^{s-1}\omega)), \\
\mathrm{Res}_D^{(s)} : & \Omega(-sD+A) & \rightarrow & \mathbb{F}^{sn} \\
& \omega & \mapsto & (\mathrm{Res}_{P_1}^{(s)}(\omega), \ldots, \mathrm{Res}_{P_n}^{(s)}(\omega))
\end{array}
$$

and $C_\Omega^{(s)}(D, A) := \mathrm{Res}_D^{(s)}(\Omega(-sD+A))$.

- If $s = 1$ then $C_L^{(s)}(D, A)^\perp$ and $C_\Omega^{(s)}(D, A)$ are dual.
- We will now show that this also holds for arbitrary s. For this it is important that the choice of local parameter t is fixed.

Duality

Proposition

We have that

1. $\dim C_L^{(s)}(D, A) = l(A) - l(-sD + A)$,
2. $C_\Omega^{(s)}(D, A) = C_L^{(s)}(D, A)^\perp$.

Duality

Proposition

We have that

1. $\dim C_L^{(s)}(D, A) = l(A) - l(-sD + A)$,
2. $C_\Omega^{(s)}(D, A) = C_L^{(s)}(D, A)^\perp$.

Proof:

- Let $g \in L(A)$. We have that $\mathrm{Ev}_D^{(s)}(g) = (0, \ldots, 0)$ if and only if g has a zero of order at least s in every $P \in \operatorname{supp} D$.
- This implies that the kernel of $\mathrm{Ev}_D^{(s)}$ is $L(-sD + A)$. This proves the first statement.

Duality

Proposition

We have that

1. $\dim C_L^{(s)}(D, A) = l(A) - l(-sD + A)$,
2. $C_\Omega^{(s)}(D, A) = C_L^{(s)}(D, A)^\perp$.

Proof:

- Let $g \in L(A)$. We have that $\mathrm{Ev}_D^{(s)}(g) = (0, \ldots, 0)$ if and only if g has a zero of order at least s in every $P \in \operatorname{supp} D$.
- This implies that the kernel of $\mathrm{Ev}_D^{(s)}$ is $L(-sD + A)$. This proves the first statement.
- For the second statement let $\omega \in \Omega(-sD + A)$ and $g \in L(A)$.

Duality

- Locally at a $P \in \operatorname{supp} D$, we can write

$$\begin{aligned} \omega &= (\beta_s t^{-s} + \cdots + \beta_1 t^{-1} + \cdots)\, dt \\ g &= \alpha_0 + \alpha_1 t + \cdots + \alpha_{s-1} t^{s-1} + \cdots \,, \end{aligned}$$

so $\mathrm{Res}_P^{(s)}(\omega) = (\beta_1, \ldots, \beta_s)$ and $\mathrm{Ev}_P^{(s)}(g) = (\alpha_0, \ldots, \alpha_{s-1})$.

Duality

- Locally at a $P \in \operatorname{supp} D$, we can write

$$\begin{aligned} \omega &= (\beta_s t^{-s} + \cdots + \beta_1 t^{-1} + \cdots)\, dt \\ g &= \alpha_0 + \alpha_1 t + \cdots + \alpha_{s-1} t^{s-1} + \cdots , \end{aligned}$$

  so $\mathrm{Res}_P^{(s)}(\omega) = (\beta_1, \ldots, \beta_s)$ and $\mathrm{Ev}_P^{(s)}(g) = (\alpha_0, \ldots, \alpha_{s-1})$.
- Then $\langle \mathrm{Res}_P^{(s)}(\omega), \mathrm{Ev}_P^{(s)}(g) \rangle$ is exactly the coefficient of t^{-1} in the product $g\omega$.
- Therefore we have

$$\langle \mathrm{Res}_P^{(s)}(\omega), \mathrm{Ev}_P^{(s)}(g) \rangle = \mathrm{res}_P(g\omega).$$

- Also note that $g\omega \in \Omega(-sD)$.

Duality

- Using all this we get

$$\langle \mathrm{Res}_D^{(s)}(\omega), \mathrm{Ev}_D^{(s)}(g) \rangle = \sum_{i=0}^{n} \mathrm{res}_{P_i}(g\omega) = 0.$$

where the last equality follows from the residue theorem.

Duality

- Using all this we get

$$\langle \mathrm{Res}_D^{(s)}(\omega), \mathrm{Ev}_D^{(s)}(g)\rangle = \sum_{i=0}^{n} \mathrm{res}_{P_i}(g\omega) = 0.$$

  where the last equality follows from the residue theorem.

- This implies that $C_\Omega^{(s)}(D,A) \subset C_L^{(s)}(D,A)^\perp$. The proposition now follows once we prove that

$$\dim C_\Omega^{(s)}(D,A) + \dim C_L^{(s)}(D,A) = sn.$$

Duality

- Using all this we get

$$\langle \mathrm{Res}_D^{(s)}(\omega), \mathrm{Ev}_D^{(s)}(g) \rangle = \sum_{i=0}^{n} \mathrm{res}_{P_i}(g\omega) = 0.$$

  where the last equality follows from the residue theorem.

- This implies that $C_\Omega^{(s)}(D, A) \subset C_L^{(s)}(D, A)^\perp$. The proposition now follows once we prove that

$$\dim C_\Omega^{(s)}(D, A) + \dim C_L^{(s)}(D, A) = sn.$$

- Similarly to the first statement, one can prove that $\dim C_\Omega^{(s)}(D, A) = \dim \Omega(-sD + A) - \dim \Omega(A)$.

Duality

- Therefore:

$$\begin{aligned}\dim C_L^{(s)}(D,A) &+ \dim C_\Omega^{(s)}(D,A)\\ &= l(A) - l(-sD+A) + \dim\Omega(-sD+A) - \dim\Omega(A)\\ &= \deg(A) - \deg(-sD+A) = sn.\end{aligned}$$

Where the second equality follows from Riemann-Roch's theorem. □

Recall that $l_i = l(A - iG)$. Also define $m_i := l(A - iG - (s-i)D)$. Then:

$$\operatorname{rank} \mathbf{M}_i^{(0)} = \dim C_L^{(s-i)}(A - iG) = l_i - m_i. \qquad (26)$$

If $\deg A < sn$ then $\dim C_L^{(s)}(D,A) = l(A)$. This is always the case in the setup of the list decoding algorithm.

A dual matrix

We can now describe the analogue of the matrix $\mathbf{H}$ from before.

Definition

- Let A and G be divisors as before, and b an integer s.t. $0 \leq b \leq s-1$.
- $\omega_1, \ldots, \omega_{(s-b)n}$ differential forms such that
  - $\mathrm{Res}_D^{(s-b)}(\omega_i)$ with $1 \leq i \leq \dim C_\Omega^{(s-b)}(D, A-bG)$, is a basis of $C_\Omega^{(s-b)}(D, A-bG)$
  - $\mathrm{Res}_D^{(s-b)}(\omega_1), \ldots, \mathrm{Res}_D^{(s-b)}(\omega_{(s-b)n})$ is a basis of $\mathbb{F}^{(s-b)n}$.

A dual matrix

We can now describe the analogue of the matrix $\mathbf{H}$ from before.

Definition

- Let A and G be divisors as before, and b an integer s.t. $0 \leq b \leq s-1$.
- $\omega_1, \ldots, \omega_{(s-b)n}$ differential forms such that
  - $\mathrm{Res}_D^{(s-b)}(\omega_i)$ with $1 \leq i \leq \dim C_\Omega^{(s-b)}(D, A-bG)$, is a basis of $C_\Omega^{(s-b)}(D, A-bG)$
  - $\mathrm{Res}_D^{(s-b)}(\omega_1), \ldots, \mathrm{Res}_D^{(s-b)}(\omega_{(s-b)n})$ is a basis of $\mathbb{F}^{(s-b)n}$.
- Then we define the $(s-b)n \times (s-b)n$ matrix.

$$\mathbf{H}_b := \begin{bmatrix} \mathrm{Res}_D^{(s-b)}(\omega_1) \\ \vdots \\ \mathrm{Res}_D^{(s-b)}(\omega_{(s-b)n}) \end{bmatrix}$$

An equivalent system

Definition

Also for $0 \leq b \leq s-1$ and $b \leq i \leq \lambda$, define the $(s-b)n \times l_i$ matrix

$$\mathbf{S}_i^{(i-b)} := \mathbf{H}_b \, \mathbf{D}_{i-b}^{(b)} \, \mathbf{M}_i^{(i-b)}.$$

$\mathbf{H}_b$ is regular, since its rows (by choice) is a basis of $\mathbb{F}^{(s-b)n}$.

An equivalent system

Definition

Also for $0 \le b \le s-1$ and $b \le i \le \lambda$, define the $(s-b)n \times l_i$ matrix

$$\mathbf{S}_i^{(i-b)} := \mathbf{H}_b \, \mathbf{D}_{i-b}^{(b)} \, \mathbf{M}_i^{(i-b)}.$$

$\mathbf{H}_b$ is regular, since its rows (by choice) is a basis of $\mathbb{F}^{(s-b)n}$.

Proposition

The interpolation equations (24) are row equivalent to the system

$$\left[\begin{array}{c|c|c|c|c|c}
\mathbf{S}_0^{(0)} & \mathbf{S}_1^{(1)} & \cdots & \mathbf{S}_{s-1}^{(s-1)} & \cdots & \mathbf{S}_\lambda^{(\lambda)} \\ \hline
\mathbf{0} & \mathbf{S}_1^{(0)} & \cdots & \mathbf{S}_{s-1}^{(s-2)} & \cdots & \mathbf{S}_\lambda^{(\lambda-1)} \\ \hline
\vdots & \ddots & \ddots & \vdots & & \vdots \\ \hline
\mathbf{0} & \cdots & \mathbf{0} & \mathbf{S}_{s-1}^{(0)} & \cdots & \mathbf{S}_\lambda^{(\lambda-s+1)}
\end{array}\right]
\begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_\lambda \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

An equivalent system

Proof:
The proposition follows after multiplying the b-th row of matrices in the matrix $\mathbf{M}$ (from the beginning of this section) with $\mathbf{H}_b$. $\square$

An equivalent system

Proof:
The proposition follows after multiplying the b-th row of matrices in the matrix $\mathbf{M}$ (from the beginning of this section) with $\mathbf{H}_b$. $\square$

- The matrices $\mathbf{S}_0^{(0)}, \ldots, \mathbf{S}_{s-1}^{(0)}$ are independent of the received word.

An equivalent system

Proof:
The proposition follows after multiplying the b-th row of matrices in the matrix $\mathbf{M}$ (from the beginning of this section) with $\mathbf{H}_b$. □

- The matrices $\mathbf{S}_0^{(0)}, \ldots, \mathbf{S}_{s-1}^{(0)}$ are independent of the received word.
- We have
$$\operatorname{rank} \mathbf{S}_i^{(0)} = l_i - m_i,$$
if $l_i < (s-i)n$, this reduces to $\operatorname{rank} \mathbf{S}_i^{(0)} = l_i$.
- If $l_i < (s-i)n$, then $\mathbf{S}_i^{(0)}$ can be written
$$\mathbf{S}_i^{(0)} = \left(\frac{\mathbf{0}}{\mathbf{B}_i^{(0)}} \right),$$
where $\mathbf{0}$ is the $(s-i)n - l_i \times l_i$ zero matrix.

Eliminating variables

- The $l_i \times l_i$ matrix $\mathbf{B}_i^{(0)}$ is regular, and thus in Gaussian elimination, we can eliminate the variables $q_{i1}, \ldots, q_{il_i}$ in all rows other than those of $\mathbf{B}_i^{(0)}$.
- For $i = 0$ the situation is very simple, since the only rows in which the variables $q_{01}, \ldots, q_{0l_0}$ occur, are the rows coming from $\mathbf{B}_0^{(0)}$.

Eliminating variables

- The $l_i \times l_i$ matrix $\mathbf{B}_i^{(0)}$ is regular, and thus in Gaussian elimination, we can eliminate the variables $q_{i1}, \ldots, q_{il_i}$ in all rows other than those of $\mathbf{B}_i^{(0)}$.
- For $i = 0$ the situation is very simple, since the only rows in which the variables $q_{01}, \ldots, q_{0l_0}$ occur, are the rows coming from $\mathbf{B}_0^{(0)}$.
- If $l_i \geq (s - i)n$, then we can eliminate $\operatorname{rank} \mathbf{S}_i^{(0)} = l_i - m_i$ variables among $q_{i1}, \ldots, q_{il_i}$.
- All in all, we can simplify the system in the proposition by eliminating $\sum_{i=0}^{s}(l_i - m_i)$ variables.

Example

- This means that the remaining $\sum_{i=0}^{s} m_i + \sum_{i=s+1}^{\lambda} l_i$ variables can be found by solving

$$\sum_{i=0}^{s}(\,(s-i)n - l_i + m_i\,)$$

linear equations.

- In general this gives a significant reduction of the size of the original system.

Example

- This means that the remaining $\sum_{i=0}^{s} m_i + \sum_{i=s+1}^{\lambda} l_i$ variables can be found by solving

$$\sum_{i=0}^{s} ((s-i)n - l_i + m_i)$$

linear equations.
- In general this gives a significant reduction of the size of the original system.

Example:

- This is a continuation of the previous example about list decoding.
- Then an interpolation polynomial was found by solving a linear system of 168 equations and 171. As we have seen, we can reduce the size of the system.

Example

- First we calculate the rank of the matrices $\mathbf{S}_i^{(0)}$:

i	0	1	2	3	4	5
$\operatorname{rank} \mathbf{S}_i^{(0)}$	35	31	27	23	16	8

Thus we can eliminate 140 variables and equations, thereby reducing the system to 28 equations in 31 variables.

Example

- First we calculate the rank of the matrices $\mathbf{S}_i^{(0)}$:

i	0	1	2	3	4	5
$\operatorname{rank} \mathbf{S}_i^{(0)}$	35	31	27	23	16	8

  Thus we can eliminate 140 variables and equations, thereby reducing the system to 28 equations in 31 variables.

- We can eliminate all 116 variables q_{ij} with $0 \leq i \leq 3$ and $1 \leq j \leq l_i$, since for $i \leq 3$ we have that $l_i < (s-i)n$.
- For $i = 4$ and $i = 5$, the situation is more complicated, but all we need to do is to compute the matrices $\mathbf{S}_4^{(0)}$ and $\mathbf{S}_5^{(0)}$ explicitly.

Example

- In order to do this, we need to choose differentials as in the definition of $\mathbf{H}_b$.

Example

- In order to do this, we need to choose differentials as in the definition of $\mathbf{H}_b$.
- Given a b between 0 and s, we can choose a basis for $\Omega(-(s-b)D + A - bG)$ with the desired properties (recall $t = x_1 + x_1^4$):

$$\omega_i = \begin{cases} f_i \, dt/t^{s-b} & \text{if } 1 \leq i < (s-b)n, \\ f_{(s-b)n+1} \, dt/t^{s-b} & \text{if } i = (s-b)n. \end{cases}$$

- Using this, we can compute all matrices $\mathbf{S}_i^{(0)}$ explicitly.

Example

- In order to do this, we need to choose differentials as in the definition of $\mathbf{H}_b$.
- Given a b between 0 and s, we can choose a basis for $\Omega(-(s-b)D + A - bG)$ with the desired properties (recall $t = x_1 + x_1^4$):

$$\omega_i = \begin{cases} f_i\, dt/t^{s-b} & \text{if } 1 \leq i < (s-b)n, \\ f_{(s-b)n+1}\, dt/t^{s-b} & \text{if } i = (s-b)n. \end{cases}$$

- Using this, we can compute all matrices $\mathbf{S}_i^{(0)}$ explicitly.
- By our choice of bases, the matrices have more structure:
  - $(\mathbf{B}_i^{(0)})_{pq} = 0$ if $p + q < l_i + 1$
  - $(\mathbf{B}_i^{(0)})_{pq} = 1$ if $p + q = l_i + 1$.
- Thus eliminating q_{ij} (with $0 \leq i \leq 3$ and $1 \leq j \leq l_i$) is easy.

Example

We find that $\mathbf{S}_4^{(0)}$ is equal to:

$$
\begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0
\end{pmatrix}.
$$

We can eliminate the 16 variables q_{4j} with $1 \leq j \leq 15$ and $j = 17$.

Example

We also find that $\mathbf{S}_5^{(0)}$ is equal to:

$$\begin{pmatrix} 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0&1&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&1&0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&1&0&0&1&0&0&0&0&0&1&0&0&1&0 \\ 1&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \end{pmatrix}.$$

Thus we can eliminate the 8 variables q_{5j} with $1 \leq j \leq 7$ and $j = 9$.

Example

- What remains is to calculate the remaining 31 variables.

Example

- What remains is to calculate the remaining 31 variables.
- Doing the elimination explicitly, we find that the vector of these remaining 31 variables is in the kernel of the 28×31 matrix:

$$\left(\begin{array}{c|c} \mathbf{A}_1 & \mathbf{A}_2 \\ \hline \mathbf{A}_3 & \mathbf{A}_4 \end{array}\right),$$

- The matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4$ are ...

Example: $\mathbf{A}_1$

$$\begin{pmatrix}
0 & 0 & 0 & 1 & \alpha & \alpha & \alpha & 0 & 0 & \alpha^2 & 1 & 0 & 1 & \alpha^2 & 1 \\
0 & 0 & \alpha^2 & \alpha & 1 & \alpha^2 & \alpha^2 & 0 & \alpha & \alpha^2 & 0 & \alpha^2 & \alpha^2 & 0 & \alpha \\
0 & \alpha^2 & 0 & \alpha^2 & 0 & 1 & 1 & \alpha & 1 & \alpha & 0 & \alpha^2 & 0 & 1 & \alpha^2 \\
0 & 0 & 0 & \alpha^2 & \alpha & 1 & \alpha & 1 & 1 & \alpha^2 & \alpha^2 & 0 & \alpha^2 & \alpha & \alpha \\
\alpha^2 & 0 & \alpha^2 & 1 & 0 & \alpha & \alpha & 1 & \alpha^2 & \alpha & \alpha^2 & 1 & \alpha^2 & 0 & 0 \\
0 & \alpha^2 & \alpha & 0 & \alpha^2 & 0 & \alpha^2 & 1 & \alpha^2 & 1 & 0 & 0 & 0 & 1 & \alpha^2 \\
0 & \alpha & 0 & \alpha & 1 & 1 & 0 & \alpha^2 & \alpha & \alpha^2 & 0 & 0 & 0 & \alpha^2 & 0 \\
\alpha^2 & 0 & 0 & 1 & 0 & \alpha^2 & 0 & 1 & \alpha & 0 & 1 & 0 & \alpha & 1 & 1 \\
\alpha & 0 & 0 & \alpha & \alpha^2 & \alpha^2 & 1 & 0 & 1 & \alpha & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha & \alpha & 1 & \alpha^2 & 0 & \alpha & 0 & \alpha^2 & 0 \\
0 & \alpha^2 & 0 & 0 & 0 & 0 & \alpha^2 & \alpha^2 & 0 & \alpha^2 & \alpha & 0 & \alpha & 1 & 1 \\
0 & 0 & \alpha & \alpha^2 & 1 & \alpha^2 & 0 & 1 & \alpha^2 & 0 & 0 & \alpha & 0 & 0 & \alpha \\
\alpha^2 & \alpha & 0 & 0 & \alpha^2 & 1 & \alpha^2 & 1 & \alpha & 1 & 0 & \alpha^2 & 0 & \alpha & \alpha^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Example: $\mathbf{A}_2$

$$
\begin{pmatrix}
0 & 1 & 0 & \alpha^2 & \alpha^2 & 0 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha^2 & \alpha & \alpha & 0 & 0 & 0 \\
1 & 0 & 0 & \alpha & \alpha^2 & 1 & \alpha^2 & \alpha^2 & 1 & \alpha & 1 & 1 & \alpha^2 & 0 & 0 & 0 \\
0 & 0 & \alpha & 0 & 1 & \alpha & \alpha & 0 & 1 & 1 & 0 & \alpha^2 & \alpha & 0 & 0 & 0 \\
0 & \alpha & 1 & \alpha^2 & \alpha & \alpha & \alpha^2 & 1 & \alpha & \alpha & 1 & 0 & 1 & 0 & 0 & 0 \\
\alpha & 0 & 1 & \alpha & 1 & 0 & 1 & \alpha & \alpha^2 & 0 & 0 & \alpha & \alpha^2 & 0 & 0 & 0 \\
\alpha & 0 & \alpha & 0 & \alpha & 0 & 1 & 0 & \alpha & 0 & 1 & 1 & 1 & 0 & 0 & \alpha \\
0 & 0 & 0 & 0 & \alpha & 0 & \alpha & 0 & \alpha & \alpha & \alpha & \alpha^2 & 0 & 0 & \alpha & 0 \\
\alpha & \alpha^2 & \alpha & \alpha & 0 & \alpha^2 & \alpha & \alpha^2 & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha & 0 & 0 & \alpha \\
\alpha^2 & 0 & \alpha^2 & 0 & \alpha^2 & \alpha^2 & 0 & \alpha^2 & 1 & \alpha & \alpha^2 & \alpha & 1 & \alpha & \alpha & 0 \\
1 & 0 & \alpha^2 & 0 & 1 & 1 & 1 & \alpha & \alpha & \alpha & 1 & 1 & \alpha^2 & 0 & \alpha & \alpha^2 \\
0 & \alpha^2 & 0 & \alpha & \alpha & 1 & \alpha^2 & \alpha & 1 & \alpha^2 & 1 & \alpha^2 & 1 & \alpha & \alpha^2 & \alpha \\
0 & 0 & \alpha & 0 & 0 & 1 & \alpha & \alpha & \alpha^2 & \alpha & \alpha & \alpha & 0 & 0 & \alpha & \alpha \\
0 & 0 & \alpha & 0 & 0 & \alpha & 0 & \alpha^2 & \alpha^2 & 0 & \alpha^2 & \alpha & \alpha^2 & \alpha^2 & \alpha & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha & 0 & 0 & \alpha^2 & 1 & 0 & 0 & 0
\end{pmatrix}
$$

Example: $\mathbf{A}_3$

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & \alpha^2 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha^2 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha & 0 & \alpha^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha^2 & 0 & 0 & \alpha & 0 & \alpha^2 & \alpha & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha^2 & \alpha & 0 & \alpha^2 & \alpha & \alpha^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha & 0 & \alpha^2 & 0 & \alpha^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha^2 & 0 & \alpha^2 & \alpha & \alpha^2 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha & 0 & \alpha & \alpha^2 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & 0 & 0 & \alpha^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & 0 & \alpha^2 & \alpha & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Example: $\mathbf{A}_4$

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha & \alpha^2 & 0 & \alpha^2 & \alpha & \alpha & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & 1 & 1 & \alpha & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & \alpha & \alpha^2 & 1 & \alpha^2 & 0 & \alpha^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha & 0 & 1 & \alpha^2 & 1 & \alpha^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha & 0 & 0 & \alpha^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha & \alpha & 0 & \alpha^2 & \alpha & 1 & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha^2 & 1 & \alpha^2 & \alpha & 0 & \alpha & 0 & 0 & 0 \\
0 & 0 & \alpha^2 & \alpha & 0 & 0 & \alpha & \alpha^2 & \alpha & \alpha^2 & 1 & 0 & \alpha^2 & 0 & 0 & 0 \\
\alpha^2 & \alpha & 0 & 0 & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & \alpha & 1 & 1 & \alpha^2 & \alpha & 0 & 0 & 0 \\
\alpha & 0 & 0 & 0 & \alpha^2 & 0 & \alpha & \alpha & 1 & \alpha^2 & \alpha & \alpha^2 & \alpha & 0 & 0 & 0 \\
0 & \alpha^2 & \alpha & \alpha^2 & 0 & \alpha & \alpha^2 & \alpha^2 & \alpha & 1 & \alpha^2 & 1 & \alpha^2 & 0 & 0 & 0 \\
0 & \alpha & \alpha^2 & 0 & \alpha & 0 & \alpha & \alpha & 1 & \alpha^2 & \alpha & \alpha^2 & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 & \alpha^2 & \alpha & 0 & 0 & 0
\end{pmatrix}$$

A solution

- This matrix is much easier to handle than the original 168×171 matrix!

A solution

- This matrix is much easier to handle than the original 168×171 matrix!
- Its kernel is 5-dimensional and one of the solutions is given by (only nonzero values are stated):

q_{58}	q_{510}	q_{511}	q_{61}	q_{62}	q_{63}
1	α^2	α	1	α^2	α

q_{64}	q_{65}	q_{66}	q_{67}	q_{71}	q_{81}	q_{82}
α^2	α^2	1	α^2	1	α^2	α

A solution

- This matrix is much easier to handle than the original 168×171 matrix!
- Its kernel is 5-dimensional and one of the solutions is given by (only nonzero values are stated):

q_{58}	q_{510}	q_{511}	q_{61}	q_{62}	q_{63}
1	α^2	α	1	α^2	α

q_{64}	q_{65}	q_{66}	q_{67}	q_{71}	q_{81}	q_{82}
α^2	α^2	1	α^2	1	α^2	α

- Setting in these values in syndrome equation system from the proposition, we can then calculate the remaining 140 variables immediately.
- This was in fact how the interpolation polynomial $Q(y)$ in the list decoding example was computed.