Fast list decoding of algebraic geometry codes

Peter Beelen[1], Johan Rosenkilde[2] and Grigory Solomatov[3] (grigorys93@gmail.com)

[1]Technical University of Denmark

[2]GitHub

[3]Tel Aviv University

Inria 08.03.2022

DTU Compute
Department of Applied Mathematics and Computer Science

Outline

- Preliminaries
  - Reed-Solomon codes
  - Guruswami-Sudan for RS codes
  - Algebraic geometry codes
- Guruswami-Sudan through holomorphy rings
  - Intuition
  - Generalizing $\deg(\cdot)$ to $\delta(\cdot)$
  - Guruswami-Sudan for AG codes
  - Computer representation
- A fast decoding algorithm
  - Previous work
  - Our strategy
  - Interpolation step
  - Root-finding

Outline

- Preliminaries
  - Reed-Solomon codes
  - Guruswami-Sudan for RS codes
  - Algebraic geometry codes
- Guruswami-Sudan through holomorphy rings
  - Intuition
  - Generalizing $\deg(\cdot)$ to $\delta(\cdot)$
  - Guruswami-Sudan for AG codes
  - Computer representation
- A fast decoding algorithm
  - Previous work
  - Our strategy
  - Interpolation step
  - Root-finding

Reed-Solomon codes

Let $\mathbb{F}$ be a finite field.

Setting

Sender:

- has message $f \in \mathbb{F}[x]_{<k}$
- computes codeword $\boldsymbol{c} = \big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n$, $P_1, \ldots, P_n \in \mathbb{F}$, $n \geq k$,

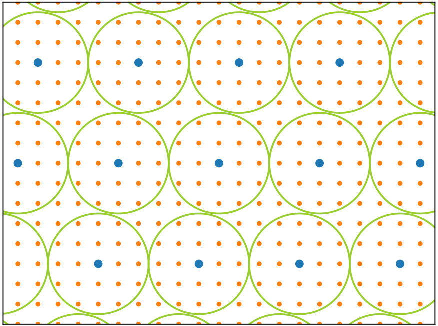
Channel:

- unknown error $\boldsymbol{e} = (e_1, \ldots, e_n) \in \mathbb{F}^n$, many $e_j = 0$
- received word $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}^n$

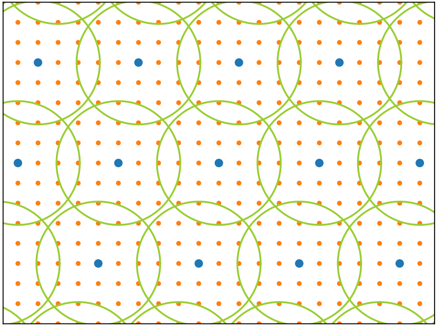
Receiver:

- Find the list containing all codewords within radius τ from $\boldsymbol{r}$.

Decoding beyond half the minimum distance



Half minimum distance



Beyond half minimum distance

Guruswami-Sudan for RS codes

Theorem

Let $s, \ell, \tau \in \mathbb{Z}_{>0}$ with $s \leq \ell$. If $Q \in \mathbb{F}[x, z]$ with $\deg_z Q \leq \ell$ satisfies

1. Q has zero of multiplicity s at every (P_j, r_j), $\quad Q = \sum_{u+v \geq s} Q_{u,v}(x - P_j)^u (z - r_j)^v$, $Q_{u,v} \in \mathbb{F}$
2. $\deg_{1,k-1} Q < s(n - \tau)$,

then $Q(x, f(x)) = 0$ whenever $d(\boldsymbol{r}, \boldsymbol{c}_f) \leq \tau$. $\quad Q(f) = 0$ if we think $Q \in \mathbb{F}[x][z]$

Guruswami-Sudan for RS codes

Theorem

Let $s, \ell, \tau \in \mathbb{Z}_{>0}$ with $s \leq \ell$. If $Q \in \mathbb{F}[x, z]$ with $\deg_z Q \leq \ell$ satisfies

1. Q has zero of multiplicity s at every (P_j, r_j), $\quad Q = \sum_{u+v \geq s} Q_{u,v}(x - P_j)^u (z - r_j)^v$, $Q_{u,v} \in \mathbb{F}$
2. $\deg_{1,k-1} Q < s(n - \tau)$,

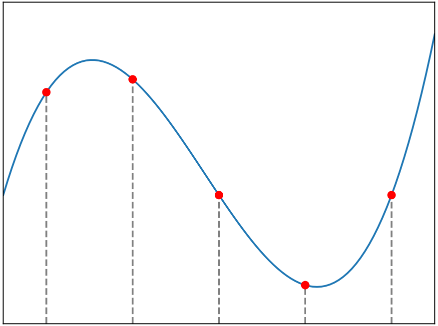
then $Q(x, f(x)) = 0$ whenever $d(\boldsymbol{r}, \boldsymbol{c}_f) \leq \tau$. $\quad Q(f) = 0$ if we think $Q \in \mathbb{F}[x][z]$

Proof: If P_j is not an error position, then $Q(P_j, f(P_j)) = Q(P_j, r_j)$.

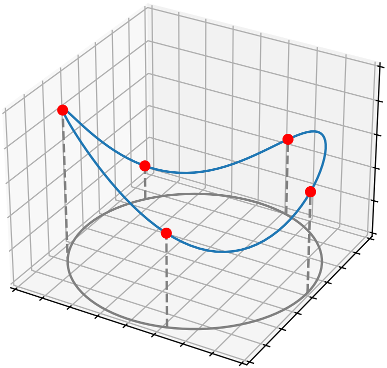
So $\widehat{Q}(x) := Q(x, f(x)) \in \mathbb{F}[x]$ has a root of multiplicity at least s at P_j, i.e. at least $s(n - \tau)$ roots in total.

But $\deg \widehat{Q}(x) < s(n - \tau)$, since $\deg f(x) \leq k - 1$, hence $\widehat{Q}(x) = 0$.

AG codes



RS codes



AG codes

Function fields

- $\mathbb{F}(x) = \{a/b \mid a, b \in \mathbb{F}[x] \text{ with } b \neq 0\}$ = rational function field
- F = finite extension of $\mathbb{F}(x) \approx$ multivariate polynomial fractions on an algebraic curve
- g = genus = number of unattainable pole orders at any point/place

Function fields

- $\mathbb{F}(x) = \{a/b \mid a, b \in \mathbb{F}[x] \text{ with } b \neq 0\}$ = rational function field
- F = finite extension of $\mathbb{F}(x) \approx$ multivariate polynomial fractions on an algebraic curve
- g = genus = number of unattainable pole orders at any point/place

Example 1 (Hermitian function field)

$F = \mathbb{F}_{q^2}(x, y)$ with $y^q + y = x^{q+1}$

$q^3 + 1$ rational places

$g = \frac{1}{2}q(q-1)$

AG codes

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\boxed{\mathcal{C}_{\mathcal{L}}(D,G) = \{\big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}}$. $(d \geq d^* := n - \deg G)$

AG codes

DTU

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\boxed{\mathcal{C}_{\mathcal{L}}(D,G) = \{\big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}}$. $(d \geq d^* := n - \deg G)$

- Place $\approx$ point on the curve $P = \langle x - \alpha \rangle_{\mathcal{O}} \approx \alpha$

AG codes

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\boxed{\mathcal{C}_{\mathcal{L}}(D, G) = \{\big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}}$. $\quad (d \geq d^* := n - \deg G)$

- Place $\approx$ point on the curve $\quad P = \langle x - \alpha \rangle_{\mathcal{O}} \approx \alpha$
- Divisor $=$ a formal sum of places $\quad P_1 - 2P_2 + 3P_3$

AG codes

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\boxed{\mathcal{C}_{\mathcal{L}}(D,G) = \{\big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}}$. $(d \geq d^* := n - \deg G)$

- Place $\approx$ point on the curve $P = \langle x - \alpha \rangle_{\mathcal{O}} \approx \alpha$
- Divisor $=$ a formal sum of places $P_1 - 2P_2 + 3P_3$
- $\deg A = \sum m_i$ for any divisor $A = \sum m_i A_i$ (for *rational* places A_i) $\deg(P_1 - 2P_2 + 3P_3) = 2$

AG codes

DTU

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\boxed{\mathcal{C}_{\mathcal{L}}(D, G) = \{\big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}}$. $(d \geq d^* := n - \deg G)$

- Place $\approx$ point on the curve $\quad P = \langle x - \alpha \rangle_{\mathcal{O}} \approx \alpha$
- Divisor $=$ a formal sum of places $\quad P_1 - 2P_2 + 3P_3$
- $\deg A = \sum m_i$ for any divisor $A = \sum m_i A_i$ (for *rational* places A_i) $\quad \deg(P_1 - 2P_2 + 3P_3) = 2$
- $v_P(a) =$ *valuation* of function $a \in F$ at place P (zero multiplicity) $\quad v_{P_{(0)}}\left(\frac{x^2}{(x-1)^3}\right) = 2$

AG codes

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\mathcal{C}_{\mathcal{L}}(D, G) = \{(f(P_1), \ldots, f(P_n)) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}$. $(d \geq d^* := n - \deg G)$

- Place $\approx$ point on the curve $\quad P = \langle x - \alpha \rangle_{\mathcal{O}} \approx \alpha$
- Divisor $=$ a formal sum of places $\quad P_1 - 2P_2 + 3P_3$
- $\deg A = \sum m_i$ for any divisor $A = \sum m_i A_i$ (for *rational* places A_i) $\quad \deg(P_1 - 2P_2 + 3P_3) = 2$
- $v_P(a) =$ *valuation* of function $a \in F$ at place P (zero multiplicity) $\quad v_{P_{(0)}}(\frac{x^2}{(x-1)^3}) = 2$
- $(a) = \sum_P v_P(a) P =$ principal divisor of a $\quad (\frac{x^2}{(x-1)^3}) = 2P_{(0)} - 3P_{(1)} + P_\infty$

AG codes

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\boxed{\mathcal{C}_{\mathcal{L}}(D,G) = \{\big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}}$. $(d \geq d^* := n - \deg G)$

- Place $\approx$ point on the curve $\quad P = \langle x - \alpha \rangle_{\mathcal{O}} \approx \alpha$
- Divisor $=$ a formal sum of places $\quad P_1 - 2P_2 + 3P_3$
- $\deg A = \sum m_i$ for any divisor $A = \sum m_i A_i$ (for *rational* places A_i) $\quad \deg(P_1 - 2P_2 + 3P_3) = 2$
- $v_P(a) =$ *valuation* of function $a \in F$ at place P (zero multiplicity) $\quad v_{P_{(0)}}\left(\frac{x^2}{(x-1)^3}\right) = 2$
- $(a) = \sum_P v_P(a) P =$ principal divisor of a $\quad \left(\frac{x^2}{(x-1)^3}\right) = 2P_{(0)} - 3P_{(1)} + P_\infty$
- $\mathcal{L}(A) = \{a \in F \setminus \{0\} \mid (a) \geq -A\} \cup \{0\} =$ *Riemann-Roch space* $\quad \mathcal{L}(3P_\infty - 2P_{(0)}) = (x^2\mathbb{F}[x])_{\deg \leq 3}$

AG codes

Definition

Given

- divisor $D = P_1 + \cdots + P_n$, where $P_1, \ldots, P_n$ are rational places (points over $\mathbb{F}$),
- divisor G with $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$,

define the code $\mathcal{C}_{\mathcal{L}}(D,G) = \{\big(f(P_1), \ldots, f(P_n)\big) \in \mathbb{F}^n \mid f \in \mathcal{L}(G)\}$. $(d \geq d^* := n - \deg G)$

- Place $\approx$ point on the curve $\quad P = \langle x - \alpha \rangle_{\mathcal{O}} \approx \alpha$
- Divisor $=$ a formal sum of places $\quad P_1 - 2P_2 + 3P_3$
- $\deg A = \sum m_i$ for any divisor $A = \sum m_i A_i$ (for *rational* places A_i) $\quad \deg(P_1 - 2P_2 + 3P_3) = 2$
- $v_P(a) =$ *valuation* of function $a \in F$ at place P (zero multiplicity) $\quad v_{P_{(0)}}(\frac{x^2}{(x-1)^3}) = 2$
- $(a) = \sum_P v_P(a)P =$ principal divisor of a $\quad (\frac{x^2}{(x-1)^3}) = 2P_{(0)} - 3P_{(1)} + P_\infty$
- $\mathcal{L}(A) = \{a \in F \setminus \{0\} \mid (a) \geq -A\} \cup \{0\} =$ *Riemann-Roch space* $\quad \mathcal{L}(3P_\infty - 2P_{(0)}) = (x^2\mathbb{F}[x])_{\deg \leq 3}$
- $l(A) := \dim_{\mathbb{F}} \mathcal{L}(A) \geq \deg A - g + 1$ (Riemann's theorem) $\quad l(3P_\infty - 2P_{(0)}) = 2$

Outline

- Preliminaries
  - Reed-Solomon codes
  - Guruswami-Sudan for RS codes
  - Algebraic geometry codes
- Guruswami-Sudan through holomorphy rings
  - Intuition
  - Generalizing $\deg(\cdot)$ to $\delta(\cdot)$
  - Guruswami-Sudan for AG codes
  - Computer representation
- A fast decoding algorithm
  - Previous work
  - Our strategy
  - Interpolation step
  - Root-finding

Intuition

- Holomorphy rings: $\mathbb{F}[x] = \mathcal{L}(\infty P_\infty)$, $\mathbb{F}[1/x] = \mathcal{L}(\infty P_{(0)})$, ... for $F = \mathbb{F}(x)$
- Like Riemann-Roch spaces, but with unrestricted pole orders.
- Infinite dimension over $\mathbb{F}$.

Generalizing $\deg(\cdot)$ to $\delta(\cdot)$

Let:

- fixed rational place $P_\infty \notin \operatorname{supp} D$, (not restrictive)
- $Я(A) = \mathcal{L}(\infty P_\infty + A) = \bigcup_{m=-\infty}^{\infty} \mathcal{L}(mP_\infty + A)$ for any divisor A, $Я := Я(0) = \mathcal{L}(\infty P_\infty)$,
- for any $a \in Я(A)$, $\delta_A(a) = -v_{P_\infty}(a) - v_{P_\infty}(A) =$ smallest m such that $a \in \mathcal{L}(mP_\infty + A)$,

  $\delta(a) := \delta_0(a) = -v_{P_\infty}(a)$.

Generalizing $\deg(\cdot)$ to $\delta(\cdot)$

Let:

- fixed rational place $P_\infty \notin \operatorname{supp} D$, (not restrictive)
- $\mathcal{H}(A) = \mathcal{L}(\infty P_\infty + A) = \bigcup_{m=-\infty}^{\infty} \mathcal{L}(mP_\infty + A)$ for any divisor A, $\mathcal{H} := \mathcal{H}(0) = \mathcal{L}(\infty P_\infty)$,
- for any $a \in \mathcal{H}(A)$, $\delta_A(a) = -v_{P_\infty}(a) - v_{P_\infty}(A) =$ smallest m such that $a \in \mathcal{L}(mP_\infty + A)$,

  $\delta(a) := \delta_0(a) = -v_{P_\infty}(a)$.

Note:

- $\delta_{A+B}(ab) = \delta_A(a) + \delta_B(b)$ for any $a \in \mathcal{H}(A)$ and $b \in \mathcal{H}(B)$,
- if $F = \mathbb{F}(x)$, then $\delta(a) = \deg a$ for any $a \in \mathcal{H} = \mathbb{F}[x]$.

Guruswami-Sudan for AG codes

Theorem (special case)

Let $s, \ell, \tau \in \mathbb{Z}_{>0}$ with $s \leq \ell$. If $Q \in F[z]$ satisfies

- $Q \in \mathcal{M}_{s,\ell}(D, G) := \Big\{ Q = \sum_{t=0}^{\ell} Q^{(t)} z^t \mid Q^{(t)} \in \mathcal{A}(-tG),$
  Q has a zero of multiplicity at least s at each $(P_j, r_j) \Big\}$,
- $\delta_G(Q) := \max_t \delta_{-tG}(Q^{(t)}) < s(n - \tau)$,

then $Q(f) = 0$.

Guruswami-Sudan for AG codes

Theorem (special case)

Let $s, \ell, \tau \in \mathbb{Z}_{>0}$ with $s \le \ell$. If $Q \in F[z]$ satisfies

- $Q \in \mathcal{M}_{s,\ell}(D,G) := \Big\{ Q = \sum_{t=0}^{\ell} Q^{(t)} z^t \mid Q^{(t)} \in \mathfrak{R}(-tG),$ Q has a zero of multiplicity at least s at each $(P_j, r_j) \Big\}$,
- $\delta_G(Q) := \max_t \delta_{-tG}(Q^{(t)}) < s(n-\tau)$,

then $Q(f) = 0$.

Proof: Since $f^t \in \mathcal{L}(tG) \subset \mathfrak{R}(tG)$, then $Q(f) = \sum_{t=0}^{\ell} Q^{(t)} f^t \in \mathfrak{R}$.

Moreover, $\delta(Q(f)) \le \max_t \{\delta_{-tG}(Q^{(t)}) + \delta(f^t)\} = \delta_G(Q) < s(n-\tau)$.

But then $Q(f) \in \mathcal{L}\Big(\underbrace{\delta(Q(f))P_\infty - s\sum_{j \notin \mathcal{E}} P_j}_{\text{negative degree}} \Big) = \{0\}$.

Computer representation

Computer representation

Fix $x \in Я = \mathcal{L}(\infty P_\infty)$ such that $\mu := \delta(x) > 0$ is minimal.

Then $Я(A)$ is $Я$-module and an $\mathbb{F}[x]$-module.

Computer representation

Fix $x \in \mathcal{H} = \mathcal{L}(\infty P_\infty)$ such that $\mu := \delta(x) > 0$ is minimal.

Then $\mathcal{H}(A)$ is $\mathcal{H}$-module and an $\mathbb{F}[x]$-module.

Definition

For any divisor A and $i = 0, \ldots, \mu - 1$, let

$$y_i^{(A)} \in \{a \in \mathcal{H}(A) \mid \delta_A(a) \equiv i \mod \mu\}$$

be such that $\delta_A(y_i^{(A)})$ is minimal. Also define $y_i := y_i^{(0)}$.

Computer representation

Fix $x \in \mathcal{H} = \mathcal{L}(\infty P_\infty)$ such that $\mu := \delta(x) > 0$ is minimal.

Then $\mathcal{H}(A)$ is $\mathcal{H}$-module and an $\mathbb{F}[x]$-module.

Definition

For any divisor A and $i = 0, \dots, \mu - 1$, let

$$y_i^{(A)} \in \{a \in \mathcal{H}(A) \mid \delta_A(a) \equiv i \mod \mu\}$$

be such that $\delta_A(y_i^{(A)})$ is minimal. Also define $y_i := y_i^{(0)}$.

Lemma

1. $y_0^{(A)}, \dots, y_{\mu-1}^{(A)}$ is an $\mathbb{F}[x]$-basis of $\mathcal{H}(A)$,
2. if $a = \sum_{i=0}^{\mu-1} a_i y_i^{(A)} \in \mathcal{H}(A)$ with $a_i \in \mathbb{F}[x]$, then $\deg a_i \leq \frac{1}{\mu}(\delta_A(a) + \deg A)$.

Outline

- Preliminaries
  - Reed-Solomon codes
  - Guruswami-Sudan for RS codes
  - Algebraic geometry codes
- Guruswami-Sudan through holomorphy rings
  - Intuition
  - Generalizing $\deg(\cdot)$ to $\delta(\cdot)$
  - Guruswami-Sudan for AG codes
  - Computer representation
- A fast decoding algorithm
  - Previous work
  - Our strategy
  - Interpolation step
  - Root-finding

Previous work

1997 **Sudan** – *Decoding RS codes beyond the error-correction bound*

1998 **Guruswami, Sudan** – *Improved decoding of RS codes and AG codes*

2010 **Beelen, Brander** – *Efficient list decoding of a class of AG codes*
Complexity: $\widetilde{\mathcal{O}}(\ell^5\mu^3(n+g))$

2015 **Rosenkilde, Beelen** – Sub-quadratic decoding on one-point Hermitian codes
Complexity: $\boxed{\widetilde{\mathcal{O}}(s\ell^\omega\mu^{\omega-1}(n+g))} = \widetilde{\mathcal{O}}(s\ell^\omega n^{(\omega+2)/3})$

2015 **Chowdhury, Jeannerod, Neiger, Schost, Villard** – *Faster algorithms for multivariate interpolation with multiplicities and simultaneous polynomial approximations* Complexity: $\widetilde{\mathcal{O}}(s^2\ell^{\omega-1}n)$

1995 **Sakata, Jensen, Høholdt** – Generalized Berlekamp-Massey decoding of AG codes up to half the Feng-Rao bound (one-point codes)
Complexity: $\mathcal{O}(\mu n^2)$.

2014 **Sakata, Fujisawa** – Fast decoding of multi-point codes from algebraic curves
Complexity: $\mathcal{O}(\mu n^2)$.

Strategy

Interpolation

Root-finding

Strategy

Interpolation

1. **Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as a $\mathcal{A}$-module.**

Root-finding

Strategy

Interpolation

1. Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as a $\mathfrak{R}$-module.

2. Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as an $\mathbb{F}[x]$-module.

Root-finding

Strategy

Interpolation

1. **Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as a $\mathcal{A}$-module.**

2. **Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as an $\mathbb{F}[x]$-module.**

3. **Find Q as a "small" element in $\mathcal{M}_{s,\ell}(D,G)$.**

Root-finding

Strategy

Interpolation

1. Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as a $\mathfrak{R}$-module.

2. Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as an $\mathbb{F}[x]$-module.

3. Find Q as a "small" element in $\mathcal{M}_{s,\ell}(D,G)$.

Root-finding

1. Compute $\mathbb{F}[\![x]\!]$-representation of the coefficients of $Q \in F[z]$.

Strategy

Interpolation

1. **Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as a $\mathcal{R}$-module.**

2. **Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as an $\mathbb{F}[x]$-module.**

3. **Find Q as a “small” element in $\mathcal{M}_{s,\ell}(D,G)$.**

Root-finding

1. **Compute $\mathbb{F}[\![x]\!]$-representation of the coefficients of $Q \in F[z]$.**

2. **Use an existing algorithm to compute $\mathbb{F}[\![x]\!]$-roots of $Q \in \mathbb{F}[\![x]\!][z]$.**

Strategy

Interpolation

1. Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as a $\mathcal{A}$-module.

2. Compute a generating set of $\mathcal{M}_{s,\ell}(D,G)$ as an $\mathbb{F}[x]$-module.

3. Find Q as a "small" element in $\mathcal{M}_{s,\ell}(D,G)$.

Root-finding

1. Compute $\mathbb{F}[\![x]\!]$-representation of the coefficients of $Q \in F[z]$.

2. Use an existing algorithm to compute $\mathbb{F}[\![x]\!]$-roots of $Q \in \mathbb{F}[\![x]\!][z]$.

3. Convert the roots to $\mathcal{A}(G)$ and filter out those that are not in $\mathcal{L}(G)$.

Module structure of $\mathcal{M}_{s,\ell}(D,G)$

Theorem

Let $R \in \mathcal{L}(G)$ such that $R(P_j) = r_j$ for $j = 1, \dots, n$.

Module structure of $\mathcal{M}_{s,\ell}(D,G)$

Theorem

Let $R \in Я(G)$ such that $R(P_j) = r_j$ for $j = 1, \ldots, n$.

1. $\mathcal{M}_{s,\ell}(D,G) = \langle \{B_v^{(u)}\}_{v=1,2}^{u=0,\ldots,\ell} \rangle_{Я}$, where $B_v^{(u)} = (z-R)^u g_v^{(u)}$ and $\langle g_1^{(u)}, g_2^{(u)} \rangle_{Я} = Я(G_u)$,

Module structure of $\mathcal{M}_{s,\ell}(D,G)$

Theorem

Let $R \in \mathcal{A}(G)$ such that $R(P_j) = r_j$ for $j = 1, \dots, n$.

1. $\mathcal{M}_{s,\ell}(D,G) = \langle \{B_v^{(u)}\}_{v=1,2}^{u=0,\dots,\ell} \rangle_{\mathcal{A}}$, where $B_v^{(u)} = (z-R)^u g_v^{(u)}$ and $\langle g_1^{(u)}, g_2^{(u)} \rangle_{\mathcal{A}} = \mathcal{A}(G_u)$,
2. $\mathcal{M}_{s,\ell}(D,G) = \langle \{y_i B_v^{(u)}\}_{v=1,2 \ i=0,\dots\mu-1}^{u=0,\dots,\ell} \rangle_{\mathbb{F}[x]}$.

Module structure of $\mathcal{M}_{s,\ell}(D,G)$

Theorem

Let $R \in \mathcal{A}(G)$ such that $R(P_j) = r_j$ for $j = 1, \ldots, n$.

1. $\mathcal{M}_{s,\ell}(D,G) = \langle \{B_v^{(u)}\}_{v=1,2}^{u=0,\ldots,\ell} \rangle_{\mathcal{A}}$, where $B_v^{(u)} = (z-R)^u g_v^{(u)}$ and $\langle g_1^{(u)}, g_2^{(u)} \rangle_{\mathcal{A}} = \mathcal{A}(G_u)$,
2. $\mathcal{M}_{s,\ell}(D,G) = \langle \{y_i B_v^{(u)}\}_{v=1,2\ i=0,\ldots\mu-1}^{u=0,\ldots,\ell} \rangle_{\mathbb{F}[x]}$.

Computation

1. Compute $B_v^{(u)} = (z-R)^u g_v^{(u)} = \sum_{t=0}^{u} \binom{u}{t} z^t (-R)^{u-r} g_v^{(u)}$ using MPE and interpolation.
2. Compute $\{y_i B_v^{(u)}\}_{v=1,2\ i=0,\ldots\mu-1}^{u=0,\ldots,\ell}$ using simultaneous Hermite-Padé approximations.
3. Construct a matrix in $\mathbb{F}[x]^{2\mu(\ell+1)\times\mu(\ell+1)}$ and compute a “small” basis (need only one small vector).

Module structure of $\mathcal{M}_{s,\ell}(D,G)$

Theorem

Let $R \in Я(G)$ such that $R(P_j) = r_j$ for $j = 1, \ldots, n$.

1. $\mathcal{M}_{s,\ell}(D,G) = \langle \{B_v^{(u)}\}_{v=1,2}^{u=0,\ldots,\ell} \rangle_{Я}$, where $B_v^{(u)} = (z-R)^u g_v^{(u)}$ and $\langle g_1^{(u)}, g_2^{(u)} \rangle_{Я} = Я(G_u)$,
2. $\mathcal{M}_{s,\ell}(D,G) = \langle \{y_i B_v^{(u)}\}_{v=1,2 \; i=0,\ldots\mu-1}^{u=0,\ldots,\ell} \rangle_{\mathbb{F}[x]}$.

Computation

1. Compute $B_v^{(u)} = (z-R)^u g_v^{(u)} = \sum_{t=0}^{u} \binom{u}{t} z^t (-R)^{u-r} g_v^{(u)}$ using MPE and interpolation.
2. Compute $\{y_i B_v^{(u)}\}_{v=1,2 \; i=0,\ldots\mu-1}^{u=0,\ldots,\ell}$ using simultaneous Hermite-Padé approximations.
3. Construct a matrix in $\mathbb{F}[x]^{2\mu(\ell+1)\times\mu(\ell+1)}$ and compute a "small" basis (need only one small vector).

Simultaneous Hermite-Padé: **Rosenkilde, Storjohann (2018)**

Multi-point evaluation

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu N + \delta_A(a) + \deg A)$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$,
- a function $a = \sum_{i=0}^{\mu-1} a_i y_i^{(A)} \in \mathcal{R}(A)$, where $a_i \in \mathbb{F}[x]$.

Output:

- evaluations $a(E_1), \ldots, a(E_N) \in \mathbb{F}$.

Multi-point evaluation

Algorithm — **Complexity:** $\widetilde{O}(\mu N + \delta_A(a) + \deg A)$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$,
- a function $a = \sum_{i=0}^{\mu-1} a_i y_i^{(A)} \in \mathfrak{R}(A)$, where $a_i \in \mathbb{F}[x]$.

Output:

- evaluations $a(E_1), \ldots, a(E_N) \in \mathbb{F}$.

1. Compute $a_i(x(E_1)), \ldots, a_i(x(E_N))$ for $i = 0, \ldots, \mu - 1$. $\triangleright$ $\deg a_i \leq \frac{1}{\mu}(\delta_A(a) + \deg A)$

Multi-point evaluation

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu N + \delta_A(a) + \deg A)$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$,
- a function $a = \sum_{i=0}^{\mu-1} a_i y_i^{(A)} \in \mathcal{R}(A)$, where $a_i \in \mathbb{F}[x]$.

Output:

- evaluations $a(E_1), \ldots, a(E_N) \in \mathbb{F}$.

1. Compute $a_i(x(E_1)), \ldots, a_i(x(E_N))$ for $i = 0, \ldots, \mu - 1$. $\triangleright$ $\deg a_i \leq \frac{1}{\mu}(\delta_A(a) + \deg A)$
2. Return $a(E_j) = \sum_{i=0}^{\mu-1} a_i(x(E_j)) y_i^{(A)}(E_j)$ for $j = 1, \ldots, N$.

Interpolation

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N+g))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $w_1, \ldots, w_N \in \mathbb{F}$.

Output:

- $a \in \mathcal{R}(A)$ such that $a(E_j) = w_j$ for $j = 1, \ldots, N$ and $\delta_A(a) \leq N + 2g - 1 - \deg A$ is minimal.

Interpolation

Algorithm — **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N+g))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $w_1, \ldots, w_N \in \mathbb{F}$.

Output:

- $a \in \mathcal{R}(A)$ such that $a(E_j) = w_j$ for $j = 1, \ldots, N$ and $\delta_A(a) \leq N + 2g - 1 - \deg A$ is minimal.

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k

Interpolation

Algorithm — **Complexity:** $\widetilde{O}(\mu^{\omega-1}(N+g))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $w_1, \ldots, w_N \in \mathbb{F}$.

Output:

- $a \in \mathfrak{R}(A)$ such that $a(E_j) = w_j$ for $j = 1, \ldots, N$ and $\delta_A(a) \leq N + 2g - 1 - \deg A$ is minimal.

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k
2. For $k = 1, \ldots, \mu$, compute $W_k, Y_{i,k} \in \mathbb{F}[x]$ for $i = 0, \ldots, \mu - 1$ such that for each $E_j \in U_k$
   $W_k(x(E_j)) = w_j$ and $Y_{i,k}(x(E_j)) = y_i^{(A)}(E_j)$

Interpolation

Algorithm — **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N+g))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $w_1, \ldots, w_N \in \mathbb{F}$.

Output:

- $a \in \mathfrak{R}(A)$ such that $a(E_j) = w_j$ for $j = 1, \ldots, N$ and $\delta_A(a) \leq N + 2g - 1 - \deg A$ is minimal.

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k
2. For $k = 1, \ldots, \mu$, compute $W_k, Y_{i,k} \in \mathbb{F}[x]$ for $i = 0, \ldots, \mu - 1$ such that for each $E_j \in U_k$ $W_k(x(E_j)) = w_j$ and $Y_{i,k}(x(E_j)) = y_i^{(A)}(E_j)$
3. Compute $a_0, \ldots, a_{\mu-1} \in \mathbb{F}[x]$ with certain degree constraints such that for $k = 1, \ldots, \mu$ $\sum_{i=0}^{\mu-1} a_i Y_{i,k} \equiv W_k \mod \prod_{E_j \in U_k}(x - x(E_j))$.

Interpolation

Algorithm — **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N+g))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $w_1, \ldots, w_N \in \mathbb{F}$.

Output:

- $a \in \mathcal{R}(A)$ such that $a(E_j) = w_j$ for $j = 1, \ldots, N$ and $\delta_A(a) \leq N + 2g - 1 - \deg A$ is minimal.

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k
2. For $k = 1, \ldots, \mu$, compute $W_k, Y_{i,k} \in \mathbb{F}[x]$ for $i = 0, \ldots, \mu - 1$ such that for each $E_j \in U_k$ $W_k(x(E_j)) = w_j$ and $Y_{i,k}(x(E_j)) = y_i^{(A)}(E_j)$
3. Compute $a_0, \ldots, a_{\mu-1} \in \mathbb{F}[x]$ with certain degree constraints such that for $k = 1, \ldots, \mu$ $\sum_{i=0}^{\mu-1} a_i Y_{i,k} \equiv W_k \mod \prod_{E_j \in U_k}(x - x(E_j))$.
4. Return $a = \sum_{i=0}^{\mu-1} a_i y_i^{(A)}$.

Basis products

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N + |\deg A|))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $a \in Я(A)$.

Output:

- products $y_0 a, \ldots, y_{\mu-1} a \in Я(a)$. $\mathbb{F}[x]$-basis of $\langle a \rangle_Я$ or $\langle z - a \rangle_Я$

Basis products

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N + |\deg A|))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $a \in Я(A)$.

Output:

- products $y_0 a, \ldots, y_{\mu-1} a \in Я(a)$. $\mathbb{F}[x]$-basis of $\langle a \rangle_Я$ or $\langle z - a \rangle_Я$

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k.

Basis products

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N + |\deg A|))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $a \in Я(A)$.

Output:

- products $y_0 a, \ldots, y_{\mu-1} a \in Я(a)$. $\mathbb{F}[x]$-basis of $\langle a \rangle_Я$ or $\langle z - a \rangle_Я$

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k.

2. For $k = 1, \ldots, \mu$, compute $Y_{i,k}, A_{i,k} \in \mathbb{F}[x]$ for $i = 0, \ldots, \mu - 1$ such that for each $E_j \in U_k$ $Y_{i,k}(x(E_j)) = y_i^{(A)}(E_j)$ and $A_{i,k}(x(E_j)) = a(E_j) y_i(E_j)$.

Basis products

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N + |\deg A|))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $a \in Я(A)$.

Output:

- products $y_0 a, \ldots, y_{\mu-1} a \in Я(a)$. $\mathbb{F}[x]$-basis of $\langle a \rangle_Я$ or $\langle z - a \rangle_Я$

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k.
2. For $k = 1, \ldots, \mu$, compute $Y_{i,k}, A_{i,k} \in \mathbb{F}[x]$ for $i = 0, \ldots, \mu - 1$ such that for each $E_j \in U_k$ $Y_{i,k}(x(E_j)) = y_i^{(A)}(E_j)$ and $A_{i,k}(x(E_j)) = a(E_j) y_i(E_j)$.
3. Compute a (shifted) Popov basis $\boldsymbol{P} = [\boldsymbol{P}_1 | \boldsymbol{P}_2] \in \mathbb{F}[x]^{2\mu \times 2\mu}$ of $\{(f_0, \ldots, f_{\mu-1}, h_0, \ldots, h_{\mu-1}) \in \mathbb{F}[x]^{2\mu} \mid \sum_{i=0}^{\mu-1} f_i Y_{i,k} \equiv \sum_{i=0}^{\mu-1} h_i A_{i,k} \mod \prod_{E_j \in U_k} (x - x(E_j))\}$

Basis products

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(\mu^{\omega-1}(N + |\deg A|))$

Input:

- divisors A and $E = E_1 + \cdots + E_N$ such that $\operatorname{supp} E \cap (\operatorname{supp} A \cup \{P_\infty\}) = \emptyset$, $a \in Я(A)$.

Output:

- products $y_0 a, \ldots, y_{\mu-1} a \in Я(a)$. $\mathbb{F}[x]$-basis of $\langle a \rangle_Я$ or $\langle z - a \rangle_Я$

1. Partition E into equally sized $U_1, \ldots, U_\mu$ such that x-coordinates don't repeat within each U_k.
2. For $k = 1, \ldots, \mu$, compute $Y_{i,k}, A_{i,k} \in \mathbb{F}[x]$ for $i = 0, \ldots, \mu - 1$ such that for each $E_j \in U_k$ $Y_{i,k}(x(E_j)) = y_i^{(A)}(E_j)$ and $A_{i,k}(x(E_j)) = a(E_j) y_i(E_j)$.
3. Compute a (shifted) Popov basis $\boldsymbol{P} = [\boldsymbol{P}_1 | \boldsymbol{P}_2] \in \mathbb{F}[x]^{2\mu \times 2\mu}$ of $\{(f_0, \ldots, f_{\mu-1}, h_0, \ldots, h_{\mu-1}) \in \mathbb{F}[x]^{2\mu} \mid \sum_{i=0}^{\mu-1} f_i Y_{i,k} \equiv \sum_{i=0}^{\mu-1} h_i A_{i,k} \mod \prod_{E_j \in U_k} (x - x(E_j))\}$
4. Return $y_m a = \sum_{i=0}^{\mu-1} f_i^{(m)} y_i^{(A)}$, where $(f_0^{(m)}, \ldots, f_{\mu-1}^{(m)})$ is the m-th of the μ smallest rows in $\boldsymbol{P}_1$.

Root-finding step

Task

Given $Q \in \mathcal{M}_{s,\ell}(D, G)$, compute all $f \in \mathcal{L}(G)$ such that $Q(f) = 0$.

Root-finding step

Task

Given $Q \in \mathcal{M}_{s,\ell}(D,G)$, compute all $f \in \mathcal{L}(G)$ such that $Q(f) = 0$.

Let:

- fixed rational place $P_0 \notin \operatorname{supp} G \cup \{P_\infty\}$ having x as a local parameter,
- for any $a \in \mathfrak{R}(A)$ with $v_{P_0}(a) \geq 0$, let $\widehat{a} \in \mathbb{F}[\![x]\!]$ be the P_0-adic power series expansion of a at P_0,
- for any $Q = \sum_{t=0}^{\ell} Q^{(t)} z^t \in \mathcal{M}_{s,\ell}(D,G)$, let $\widehat{Q} = \sum_{t=0}^{\ell} \widehat{Q}^{(t)} z^t$.

Root-finding step

Task

Given $Q \in \mathcal{M}_{s,\ell}(D,G)$, compute all $f \in \mathcal{L}(G)$ such that $Q(f) = 0$.

Let:

- fixed rational place $P_0 \notin \operatorname{supp} G \cup \{P_\infty\}$ having x as a local parameter,
- for any $a \in \mathcal{A}(A)$ with $v_{P_0}(a) \geq 0$, let $\widehat{a} \in \mathbb{F}[\![x]\!]$ be the P_0-adic power series expansion of a at P_0,
- for any $Q = \sum_{t=0}^{\ell} Q^{(t)} z^t \in \mathcal{M}_{s,\ell}(D,G)$, let $\widehat{Q} = \sum_{t=0}^{\ell} \widehat{Q}^{(t)} z^t$.

Strategy **Complexity:** $\widetilde{\mathcal{O}}(\ell^2 \mu^{\omega-1}(n+g))$

1. Compute $\widehat{Q} = \sum_{t=0}^{\ell} \widehat{Q}^{(t)} z^t$. Writing $Q^{(t)} = \sum_{i=0}^{\mu-1} Q^{(t)} y_i^{(G_t)}$, then $\widehat{Q^{(t)}} = \sum_{t=0}^{\ell} Q^{(t)} \widehat{y}_i^{(G_t)}$.
2. Compute $\mathbb{F}[\![x]\!]$-roots of $\widehat{Q}$ to precision $\beta \geq 2\ell \deg G + s(n-\tau)$.
3. Convert these roots back to $\mathcal{A}(G)$ and discard those that are not in $\mathcal{L}(G)$. (dominates)

Root-finding step

Task

Given $Q \in \mathcal{M}_{s,\ell}(D,G)$, compute all $f \in \mathcal{L}(G)$ such that $Q(f) = 0$.

Let:

- fixed rational place $P_0 \notin \operatorname{supp} G \cup \{P_\infty\}$ having x as a local parameter,
- for any $a \in \mathcal{A}(A)$ with $v_{P_0}(a) \geq 0$, let $\widehat{a} \in \mathbb{F}[\![x]\!]$ be the P_0-adic power series expansion of a at P_0,
- for any $Q = \sum_{t=0}^{\ell} Q^{(t)} z^t \in \mathcal{M}_{s,\ell}(D,G)$, let $\widehat{Q} = \sum_{t=0}^{\ell} \widehat{Q}^{(t)} z^t$.

Strategy **Complexity:** $\widetilde{\mathcal{O}}(\ell^2 \mu^{\omega-1}(n+g))$

1. Compute $\widehat{Q} = \sum_{t=0}^{\ell} \widehat{Q}^{(t)} z^t$. Writing $Q^{(t)} = \sum_{i=0}^{\mu-1} Q^{(t)} y_i^{(G_t)}$, then $\widehat{Q^{(t)}} = \sum_{t=0}^{\ell} Q^{(t)} \widehat{y}_i^{(G_t)}$.
2. Compute $\mathbb{F}[\![x]\!]$-roots of $\widehat{Q}$ to precision $\beta \geq 2\ell \deg G + s(n-\tau)$.
3. Convert these roots back to $\mathcal{A}(G)$ and discard those that are not in $\mathcal{L}(G)$. (dominates)

Root-finding over $\mathbb{F}[\![x]\!]$: **Neiger, Rosenkilde, Schost (2017)**

Converting $\mathbb{F}[\![x]\!]$-roots to $\mathcal{L}(G)$-roots

Lemma

For any $\alpha > \deg G$, if

- $f \in \mathcal{L}(G)$,
- $\sum_{i=0}^{\mu-1} f_i \widehat{y}_i^{(G)} \equiv \widehat{f} \pmod{x^\alpha}$ for some $f_i \in \mathbb{F}[x]$ with $\deg f_i \leq -\frac{1}{\mu}\delta_G(y_i^{(G)})$,

then $\sum_{i=0}^{\mu-1} f_i y_i^{(G)} = f$.

Converting $\mathbb{F}[\![x]\!]$-roots to $\mathcal{L}(G)$-roots

Lemma

For any $\alpha > \deg G$, if

- $f \in \mathcal{L}(G)$,
- $\sum_{i=0}^{\mu-1} f_i \widehat{y}_i^{(G)} \equiv \widehat{f} \pmod{x^\alpha}$ for some $f_i \in \mathbb{F}[x]$ with $\deg f_i \leq -\frac{1}{\mu}\delta_G(y_i^{(G)})$,

then $\sum_{i=0}^{\mu-1} f_i y_i^{(G)} = f$.

Proof: Since $h := \sum_{i=0}^{\mu-1} f_i y_i^{(G)} \in \mathcal{L}(G) \cap (\widehat{f} + x^\alpha \mathbb{F}[\![x]\!])$, then $h - f \in \mathcal{L}(G - \alpha P_0) = \{0\}$.

Complete decoder

Algorithm **Complexity:** $\widetilde{\mathcal{O}}(s\ell^\omega \mu^{\omega-1}(n+g))$

1. Compute $B_v^{(u)} = \sum_{t=0}^{u} \binom{u}{t} z^t (-R)^{u-r} g_v^{(u)}$ for $u = 0, \ldots, \ell$ and $v = 1, 2$
2. Compute $\{y_i B_v^{(u)}\}_{v=1,2 \ i=0,\ldots,\mu-1}^{u=0,\ldots,\ell}$
3. Construct a matrix in $\mathbb{F}[x]^{2\mu(\ell+1)\times\mu(\ell+1)}$ and compute its shifted Popov form $\boldsymbol{P} \in \mathbb{F}[x]^{\mu(\ell+1)\times\mu(\ell+1)}$
4. Extract $Q \in \mathcal{M}_{s,\ell}(D, G)$ with $\delta_G(Q) < s(n-\tau)$ from $\boldsymbol{P}$
5. Compute $\widehat{Q} \in \mathbb{F}[\![x]\!][z]$ and its $\mathbb{F}[\![x]\!]$-roots
6. Convert the roots to $\mathcal{A}(G)$, discarding those that are not in $\mathcal{L}(G)$ or are far from $\boldsymbol{r}$

Conclusion

Results:

- Can list decode *any* AG code with cost $\widetilde{\mathcal{O}}(s\ell^{\omega}\mu^{\omega-1}(n+g))$.
- Faster than any other general list decoding algorithm.
- At least as fast as any specialized algorithm. (except for RS codes)

Future:

- Can we get $\widetilde{\mathcal{O}}(s^2\ell^{\omega-1}\mu^{\omega-1}(n+g))$?

2015 **Chowdhury, Jeannerod, Neiger, Schost, Villard** – *Faster algorithms for multivariate interpolation with multiplicities and simultaneous polynomial approximations*
Complexity: $\widetilde{\mathcal{O}}(s^2\ell^{\omega-1}n)$

Popov forms of matrices over $\mathbb{F}[x]$

Definition

- Pivot of a row: rightmost entry of maximal degree.
- **Popov form:** all pivots lie on the diagonal, are monic and dominate their colums.

$$\begin{pmatrix} \boxed{x^3} & x^1 & x^2 & x^0 \\ x^2 & \boxed{x^2} & x^1 & x^0 \\ x^4 & x^1 & \boxed{x^4} & x^0 \\ x^1 & x^1 & x^1 & \boxed{x^1} \end{pmatrix}$$

Properties:

- for any $\boldsymbol{M} \in \mathbb{F}[x]^{m \times m}$ there is a unique $\boldsymbol{P} \in \mathbb{F}[x]^{m \times m}$ in Popov form with the same row space,
- $\boldsymbol{P}$ has minimal row-degrees,
- can compute $\boldsymbol{P}$ with cost $\widetilde{\mathcal{O}}(m^\omega \deg \boldsymbol{M})$. (2017 **Neiger, Xuan**)