Beyond fast multiplication in $\bar{\mathbb{F}}_p$

the Artin-Schreier component

Luca De Feo

Department of Combinatorics and Optimization

August 5, 2011,
Symbolic Computation Group, University of Waterloo

From my Magma 2.11 console

```
> K<x> := GF(2,10);
> L<y> := GF(2,6);
> x+y;
$.1^29 + $.1^26 + $.1^25 + $.1^24 + $.1^22 + $.1^20 + $.1^18 +
    $.1^17 + $.1^14 + $.1^11 + $.1^10 + $.1^9 + $.1^6 + $.1^5 + 1
> Parent(x+y);
Finite field of size 2^30
> Trace(x^5*y, GF(2,2));
$.1
```

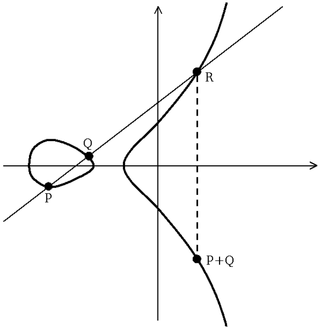
Why are these computations important?

- Geometrical algorithms over finite fields;
- Computations with number fields;
- Computations with extensions of $\mathbb{Q}_p$.

Nota: Sage is working its way to it.

An example from number theory

Elliptic curve: set of solutions in $\mathbb{P}^2(\bar{k})$ to $Y^2Z = X^3 + aXZ^2 + bZ^3, \ a, b \in k$.



Geometric addition law: given by algebraic formulas

Multiplication: write $[m]P$ for $\underbrace{P + P + \cdots + P}_{m \text{ times}}$

Torsion: $E[m] \cong (\mathbb{Z}/m\mathbb{Z})^2$, its elements are solutions to algebraic equations of degree $\sim m^2$.

$$\begin{tikzcd}[column sep=small]
 & k(L \cup K) \arrow[dl, dash] \arrow[dr, dash] & \\
L \arrow[dr, dash] & & K \arrow[dl, dash] \\
 & k &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & E[n] \oplus E[m] \arrow[dl, dash] \arrow[dr, dash] & \\
E[n] \arrow[dr, dash] & & E[m] \arrow[dl, dash] \\
 & \{0\} &
\end{tikzcd}$$

An example from number theory (cont'd)

$$
\begin{tikzcd}
\mathbb{F}_{p^{p^2}} \arrow[d, dash] & E[p^2] \arrow[r] \arrow[d, dash] & E'[p^2] \arrow[d, dash] \\
\mathbb{F}_{p^p} \arrow[d, dash] & E[p] \arrow[r] \arrow[d, dash] & E'[p] \arrow[d, dash] \\
\mathbb{F}_p & \{0_E\} \arrow[r] & \{0_{E'}\}
\end{tikzcd}
$$

Couveignes 1996 algorithm

Input: Curves E, E' over $\mathbb{F}_p$,
Output: An algebraic morphism $E \to E'$.

- Compute $E[p^k]$ and $E'[p^k]$,
- Interpolate the algebraic map from $E[p^k]$ to $E'[p^k]$.

Complexity issues

- Number of operations in $\bar{\mathbb{F}}_p$ quadratic in the degree of the map.
- But what information does this really give on the actual running time?
- De Feo 2010 shows that the number of operations in $\mathbb{F}_p$ can be made cubic. (It also improves this to quadratic, but we won't talk about this today)

So, what is an asymptotically good way of constructing $\bar{\mathbb{F}}_p$?

Constructing $\bar{\mathbb{F}}_p$

$$\bar{\mathbb{F}}_p \text{ is the inductive limit } \varinjlim_{n>0} \mathbb{F}_{p^n}$$

Compatibility: Fix embeddings such that $\mathbb{F}_{p^\ell} \subset \mathbb{F}_{p^m} \subset \mathbb{F}_{p^n}$ whenever $\ell|m|n$, then

$$\bar{\mathbb{F}}_p = \bigcup_n \mathbb{F}_{p^n}.$$

Height: Let $x \in \bar{\mathbb{F}}_p$, its *height* $h(x)$ is the degree of its minimal polynomial over $\mathbb{F}_p$; equivalently $h(x)$ is the degree of the smallest extension of $\mathbb{F}_p$ containing x.

Size: A construction is space-optimal if any $x \in \bar{\mathbb{F}}_p$ can be represented using $O(h(x) \log p)$ bits;

Arithmetic: A construction is time-optimal (resp. quasi-optimal) if any field operation on $x, y \in \bar{\mathbb{F}}_p$ can be realized in $O(\operatorname{lcm}(h(x), h(y)) \log p)$ (resp. $\tilde{O}(\operatorname{lcm}(h(x), h(y)) \log p)$) binary operations.

Beyond multiplication in $\bar{\mathbb{F}}_p$

Beware! The representation of $x \in \bar{\mathbb{F}}_p$ need not be unique.

Membership:	compute $h(x)$;
Canonical form:	compute a canonical form of size $O(h(x) \log p)$;
Traces:	$\mathrm{Tr}_{\mathbb{F}_{p^n}/\mathbb{F}_{p^m}}(x)$;
Minimal polynomials:	over $\mathbb{F}_p$, over $\mathbb{F}_{p^n}$;
Frobenius:	compute x^{p^n} using a number of operations subexponential in $\log n$;
Galois groups:	representing and realizing the action of $\mathrm{Gal}(\bar{\mathbb{F}}_p/\mathbb{F}_{p^n})$ and $\mathrm{Gal}(\mathbb{F}_{p^n}/\mathbb{F}_{p^m})$.

Classical solutions

Factorization + linear algebra (Bosma, Cannon, and Steel 1997)

- For $m|n$, construct $\mathbb{F}_{p^n}$ and $\mathbb{F}_{p^m}$ using arbitrary irreducible polynomials f_n, f_m;
- Factor f_m in $\mathbb{F}_{p^n}$, construct the embedding $\mathbb{F}_{p^m} \subset \mathbb{F}_{p^n}$ by linear algebra.

Conway polynomials (Parker 1990)

Fix the ordering $0 < 1 < \cdots < p-1$ and extend it lexicographically to $\mathbb{F}_p[-X]$. For any n, define the Conway polynomial $C_n \in \mathbb{F}_p[-X]$ as

Primitivity: C_n is primitive, i.e. any of its roots in $\bar{\mathbb{F}}_p$ generates $\mathbb{F}^*_{p^n}$;

Compatibility: For each $m|n$ and each root α of C_n, $\alpha^{(p^n-1)/(p^m-1)}$ is a root of C_m;

Uniqueness: C_n is the least antimonic polynomial satisfying these conditions.

Note: Sage drops uniqueness for large n.

Conway's $\mathbf{On}_2$

- In On Numbers and Games Conway defines surreal numbers, a very large really closed Field containing every ordinal.
- One chapter of the book is devoted to $\mathbf{On}_2$ the characteristic 2 analog of surreal numbers.
- $\mathbf{On}_2$ can be seen as the simplest way of imposing a field structure on ordinals. Starting from 0 and going upwards:
  - If the ordinal α is not a group, then $\alpha = \beta + \gamma$, where β and γ are the smallest ordinals not having a sum yet;
  - If α is not a ring, $\alpha = \beta\gamma$, where ...;
  - If α is not a field, $\alpha = \beta^{-1}$, where ...;
  - If α is not algebraically closed, α is a root of the lexicographically smallest polynomial not having a root in α;
  - If α is algebraically closed, then it is transcendental.
- This construction identifies ω with $\varinjlim \mathbb{F}_{2^{2^n}}$ and ω^{ω^ω} with $\bar{\mathbb{F}}_2$.
- The polynomials defining the successive algebraic extensions have nothing to do with Conway polynomials.

Cantor's construction of $\varinjlim \mathbb{F}_{p^{p^n}}$

Theorem (Cantor 1989)

Let $x_1, x_2 \ldots$ *be a sequence of elements in* $\bar{\mathbb{F}}_p$ *such that*

$$x_n^p - x_n = (x_1 x_2 \cdots x_{n-1})^{p-1} + [\textit{terms of lower degree}],$$

then $\mathbb{F}_p[x_n] = \mathbb{F}_{p^{p^n}}$.

- Conway's construction takes $x_n^2 - x_n = x_1 \cdots x_{n-1}$, I believe;
- Cantor suggests taking $x_n^p - x_n = x_{n-1}^{2p-1}$, because there are nice formulas to compute the minimal polynomial of x_n over $\mathbb{F}_p$.
- Cantor gives no efficient way of multiplying.

Our modest contribution (De Feo and Schost 2009)

- Simplified Cantor's proof;
- Generalized to construct $\varinjlim \mathbb{F}_{q^{p^n}}$ for any $q = p^m$;
- Given a fast multiplication algorithm and other gimmicks.

Change of representation

Triangular ideals

Cantor's construction may as well be written in terms of reduction modulo a triangular ideal:

$$\mathbb{F}_{p^{p^n}} \cong \mathbb{F}_p[X_1,\dots,X_n]/I_n \qquad \text{where}$$

$$I_n = \begin{cases} X_n^p - X_n - X_{n-1}^{2p-1} \\ \vdots \\ X_2^p - X_2 - X_1^{2p-1} \\ X_1^p - X_1 - 1 \end{cases}$$

This representation is very handy to express the embeddings.

Univariate representation

By Cantor's theorem $\mathbb{F}_p[x_n] = \mathbb{F}_{p^{p^n}}$, thus there is an univariate polynomial Q_n of degree p^n such that

$$\mathbb{F}_{p^{p^n}} \cong \mathbb{F}_p[X_n]/Q_n(X_n).$$

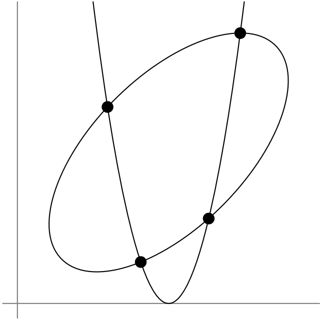
This representation is good for multiplication.

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



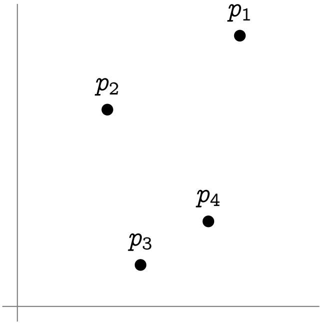
$$\begin{cases} Y + aX^2 + b \\ X^2 + cXY + dY^2 + eX + fY + g \end{cases}$$

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



A zero-dimensional ideal is just a set of points in the algebraic closure

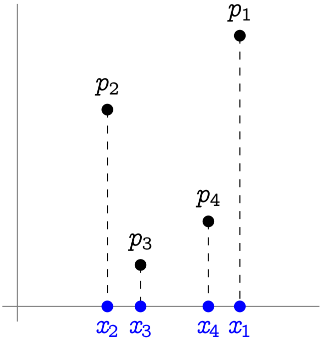
$$\begin{cases} Y + aX^2 + b \\ X^2 + cXY + dY^2 + eX + fY + g \end{cases}$$

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



Project on some separating form, compute the minimal polynomial

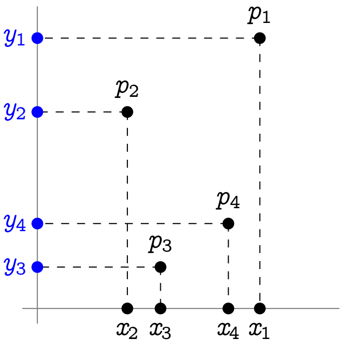
$$\left\{ \prod_{i=1}^{4}(X - x_i) \right.$$

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



Interpolate

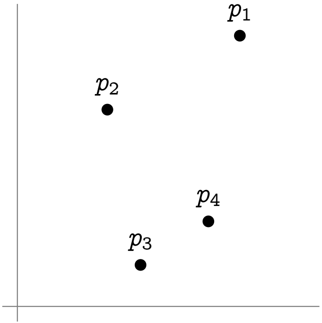
$$\begin{cases} \displaystyle\prod_{i=1}^{4}(X - x_i) \\ \displaystyle Y - \sum_{i=1}^{4} y_i \prod_{j\neq i} \frac{X - x_j}{x_i - x_j} \end{cases}$$

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \quad \cdots \quad X_n = \frac{g_n(T)}{g(T)}$$



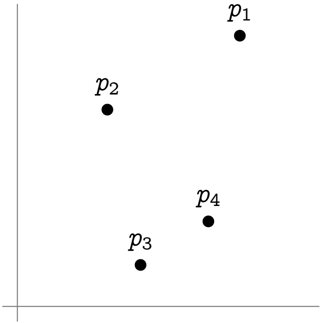
Problem: the points may not be rational

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



By the CRT

$$\bar{\mathbb{K}}[X, Y]/I \cong \bigoplus_i \bar{\mathbb{K}}[X, Y]/\mathfrak{m}_i$$

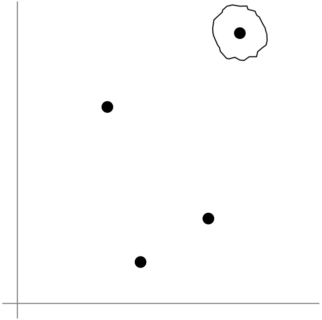
where $\mathfrak{m}_i$ is the maximal ideal at p_i.

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



Reduction modulo $\mathfrak{m}_i$ is equivalent to evaluating polynomials in $\bar{\mathbb{K}}[X, Y]$ at p_i:

$$\zeta_i : \bar{\mathbb{K}}[X, Y]/I \to \bar{\mathbb{K}}[X, Y]/\mathfrak{m}_i$$
$$a \mapsto a(p_i).$$

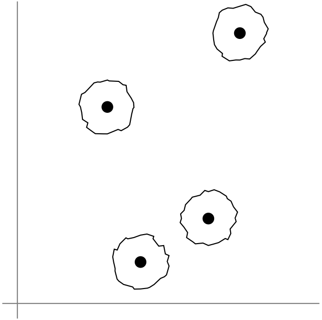
The ζ_i are linear forms on $\bar{\mathbb{K}}[X, Y]$

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



The form

$$\mathrm{Tr} = \sum_i \zeta_i$$

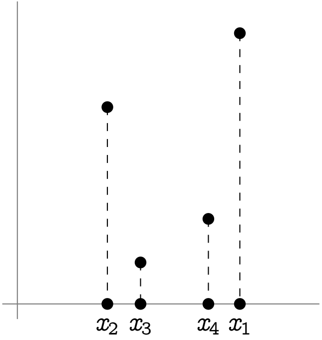
is also linear on $\bar{\mathbb{K}}[X, Y]$, but its restriction to $\mathbb{K}[X, Y]$ is $\mathbb{K}$-linear.

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



We have the formula on formal power series

$$\exp \int \sum_{j>0} \frac{\mathrm{Tr}(X^j)}{T^{j+1}} = \prod_{i=1}^{4} (T - x_i)$$

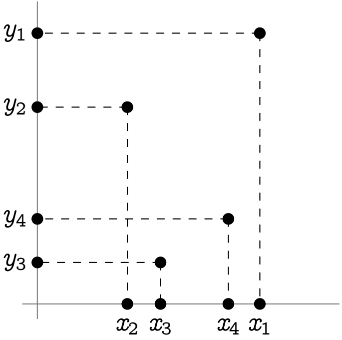
Analogous to Newton formulas.

Rational Univariate Representation (Rouillier 1999)

Goal

Express a zero-dimensional ideal in the form

$$f(T) = 0, \qquad X_1 = \frac{g_1(T)}{g(T)}, \qquad \cdots \qquad X_n = \frac{g_n(T)}{g(T)}$$



And analogously to Lagrange interpolation, we have

$$\sum_{j>0} \frac{\mathrm{Tr}(YX^j)}{T^{j+1}} = \sum_{i=1}^{4} \frac{y_i}{T - x_i}.$$

Computing many traces at once

We are left with the problem of efficiently computing

$$\sum_{j>0} \frac{\mathrm{Tr}(X^j)}{T^{j+1}}$$

Power projection (Shoup 1999; Bostan, Salvy, and Schost 2003)

Given a linear form $\ell \in (\mathbb{K}[X, Y]/I)^*$, compute $\ell(X^j)$ for many j's

$$\begin{aligned} \mathrm{proj} : (\mathbb{K}[X, Y]/I)^* &\to \mathbb{K}[[1/T]] \\ \ell &\mapsto \sum_{j>0} \frac{\ell(X^j)}{T^j} \end{aligned}$$

By taking duals:

$$\begin{aligned} \mathrm{proj}^* : \mathbb{K}[T] &\to \mathbb{K}[X, Y]/I \\ f &\mapsto f \bmod I \end{aligned}$$

and this latter problem is easy in our case, since I has a nice form.

Algebraic complexity and transposed circuits

Transposition principle (De Feo 2010; Fiduccia 1973)

From any family $(C_n)_{n\in\mathbb{N}}$ of linear arithmetic circuits with sizes $|C_n|$, one can deduce a family $(C_n^*)_{n\in\mathbb{N}}$ computing the dual problems with the sizes $|C_n^*| = |C_n|$.

- This principle carries over to straight-line programs, preserving space and time algebraic complexity.
- It carries over to more general programs, preserving time complexity and with precise bounds on space complexity.
- It can be fully automatized (De Feo and Schost 2010).
- In particular, transposing the algorithm for reduction from univariate to multivariate representation gives an algorithm for power projection with the same complexity.

Beyond multiplication in $\bar{\mathbb{F}}_p$

Summarizing, given inputs of size $p^n \log p$

Multiplication:		$\tilde{O}(p^{n+1})$
Membership:	compute $h(x)$;	$\tilde{O}(p^{n+1})$
Canonical form:	of size $O(h(x) \log p)$;	$\tilde{O}(p^{n+1})$
Traces:	$\mathrm{Tr}_{\mathbb{F}_{p^{p^n}}/\mathbb{F}_{p^{p^m}}}(x)$;	$\tilde{O}(p^{n+1})$
Minimal polynomials:	over $\mathbb{F}_p$, over $\mathbb{F}_{p^{p^m}}$;	$\tilde{O}(p^{n+1})$
Frobenius:	compute $x^{p^{p^m}}$ for $m < n$;	$\tilde{O}(p^{n+2})$
Solving Artin-Schreier equations:	isomorphisms of Artin-Schreier towers	$\tilde{O}(p^{n+2})$
Normal basis		free

Implementations

- FAAST (Fast Arithmetic in Artin-Schreier towers): `C++` with NTL implementation ($\sim$ 6000 lines) released under GPL:
  `http://www.lix.polytechnique.fr/~defeo/FAAST/`
- `C++` benchmarks of Couveignes' algorithm, built on top of FAAST.
- Including FAAST and algorithms for isogenies in Sage.
- Writing a compiler for automatic transposition
  `http://transalpyne.gforge.inria.fr/`

Implementations

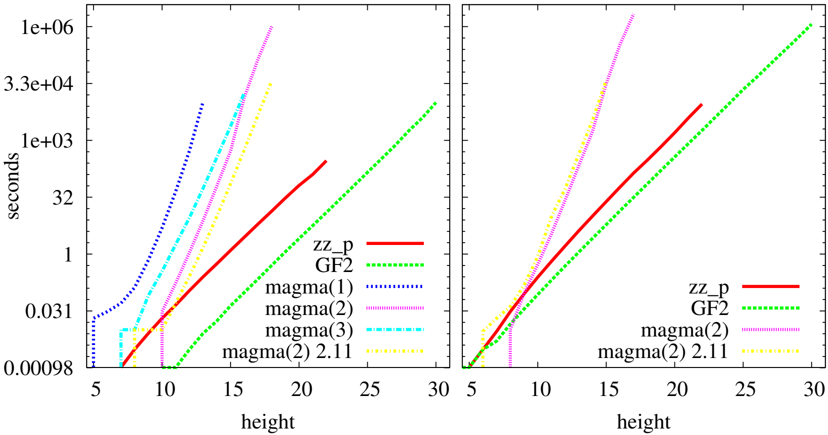


Figure: Build time (left) and isomorphism time (right) with respect to tower height. Plot is in logarithmic scale.

Fast Artin-Schreier vs Normal bases

Fast normal bases for the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$

Normal bases allow fast computation of the Frobenius morphism. Fast multiplication in such bases is an active research field.

- Low complexity normal bases $O(m^2)$
- Gauss periods (Gao, Gathen, Panario, and Shoup 2000) $\tilde{O}(m)$
- Elliptic bases (Couveignes and Lercier 2009) $\tilde{O}(m)$

Each of the above has limitations and requires a search for feasible parameters (q, m).

Fast Frobenius using Artin-Schreier towers

Restricted to $m = p^k$, but:

- Efficiency: quasi-optimal and fast in practice;
- Instantaneous: very limited precomputations, no search;
- Scalability: infinite family of parameters (q, p^k) for any k;
- Especially interesting for coding theory: $(2, 2^k)$.

Gabidulin codes

Gabidulin codes

Let $[i] \equiv q^i$, a linearized polynomial is one of the form

$$L_f = f_0 X + f_1 X^{[1]} + \cdots + f_{k-1} X^{[k-1]}.$$

An (n, k)-Gabidulin code is

$$C \quad = \quad \{(L_f(\alpha_0), \ldots, L_f(\alpha_{n-1}) \mid \deg_\sigma L_f < k\} \quad \subset \quad F_{q^m}^n.$$

Gabidulin codes are MRD. They are the rank-distance equivalent of Reed-Solomon codes.

Decoding of Gabidulin codes

Symbolic product

Given two linearized polynomials L_f, L_g, their symbolic product (or skew product) is

$$L_f \otimes L_g = L_f(L_g)$$

When L_f, L_g have coefficients in $\mathbb{F}_q$, this is equivalent to the ordinary product.

Wachter, Afanassiev, and Sidorenko 2011

- Gabidulin codes can be decoding using the linearized equivalent of the extended Euclidean algorithm;
- The complexity of the algorithm is $O(S(m) \log m)$, where $S(m)$ is the cost of performing symbolic product modulo $X^{[m]} - X$.
- Using low-complexity normal bases, $S(m) = O(m^3)$.

Fast symbolic product using low-complexity normal bases

q-transforms

- Let β be $\mathbb{F}_q$-normal. The q-transform of L_f w.r.t. β is

$$\left(L_f(\beta^{[0]}),\ldots,L_f(\beta^{[m-1]})\right);$$

- If $\tilde{\beta}$ is the dual normal element to β, the q-transform w.r.t. $\tilde{\beta}$ is the inverse q-transform w.r.t. β.

Evaluation-Interpolation strategy

- Compute $(G_0,\ldots,G_{m-1})$, the q-transform of L_g; $O(m^3)$
- Compute $H=(L_f(G_0),\ldots,L_f(G_{m-1}))$; $O(m^3)$
- Compute the inverse q-transform of H. $O(m^3)$

Faster symbolic product using Artin-Schreier towers

Key observations

- The q-transform is an ordinary modular product of polynomials;
- An $\mathbb{F}_q$-normal element is available for free in our Artin-Schreier construction;
- Computing the whole normal basis only costs $\tilde{O}(m\mathsf{M}(m))$.

Evaluation-Interpolation strategy

- Compute $(G_0, \ldots, G_{m-1})$, the q-transform of L_g; $O(\mathsf{M}(m^2))$
- Compute $H = (L_f(G_0), \ldots, L_f(G_{m-1}))$; $O(m^\omega)$
- Compute the inverse q-transform of H. $O(\mathsf{M}(m^2))$

Remarks

- Not practical, because m^ω is in practice very close to m^3;
- Similar complexities can be obtained using elliptic bases (and probably Gauss periods too).