Point counting on hyperelliptic curves of genus 3 and higher in large characteristic

Simon Abelard, Pierrick Gaudry, Pierre-Jean Spaenlehauer

Caramba – LORIA, Nancy
CNRS, Université de Lorraine, Inria

ECC 2018 – Osaka

Plan

Introduction, background

Schoof's algorithm

Tools for polynomial systems

New complexity for large genus

The case of genus 3 with real multiplication

Point counting

Let $\mathcal{C}$ be a curve of genus g over a finite field $\mathbb{F}_q$.
The number $N_{i,\mathcal{C}}$ of $\mathbb{F}_{q^i}$-rational points of $\mathcal{C}$ is finite.

The **Zeta function** collects all of them into an analytic object:

$$Z(\mathcal{C}, T) = \exp\left(\sum_{i\geq 1} N_{i,\mathcal{C}} \frac{T^i}{i}\right).$$

Weil's theorem:

$$Z(\mathcal{C}, T) = \frac{P_{\mathcal{C}}(T)}{(1-T)(1-qT)},$$

where $P_{\mathcal{C}}(T) = q^{2g}T^{2g} + \cdots$ is with integer coefficients.

Our goal: compute $P_{\mathcal{C}}(T)$ (hence, $Z(\mathcal{C}, T)$ and all of the $N_{i,\mathcal{C}}$).

Algorithmic Holy Grail

Size of the input: $O(g \log q)$

Holy Grail of point counting: find an algorithm that compute $Z_{\mathcal{C}}$

- in **polynomial time** in g and $\log q$;
- for a class of curves as large as possible;
- ...and maybe in a deterministic way;
- ...and maybe for other algebraic varieties;
- ...and maybe also in practice.

A very brief history of point counting

- 1985: Schoof's algorithm, polynomial-time, deterministic for **elliptic curves**;
- 1990: Pila, polynomial-time for **fixed genus**, deterministic for Abelian varieties (and therefore Jacobian of curves),
- 1999-20xx: Satoh, Kedlaya, Lauder-Wan, polynomial-time, deterministic in **fixed characteristic**, with p-adic algorithms.
- 2014: Harvey, **average polynomial-time** when dealing with many $\mathcal{C}$ that are reductions of the same curve over $\mathbb{Q}$.

Recent research topics

- extend p-adic techniques to more varieties (Harvey, Tuitman);
- extend average polynomial-time to more varieties (Harvey, Kedlaya, Sutherland, Massierer);
- explicit isogenies and modular equations for genus 2 (Couveignes, Ezome, Milio, Martindale);
- not so much on ℓ-adic methods

Our plan for today

Let's concentrate on **hyperelliptic curves in large characteristic**.

Known complexities for arbitrary genus:

- Pila (1990): $O(\log q)^{\Delta}$, where $\Delta(g)$ is not explicit;
- Huang, with Ierardi (1998) and Adleman (2001): $(\log q)^{\tilde{O}(g^2)}$.

First goal: make the exponent **linear** in g.

Known complexities for small genus:

- Elliptic curves: Schoof (1985), and Schoof-Elkies-Atkin (199x): $\tilde{O}((\log q)^4)$;
- Genus 2: G.-Harley (2000) and G.-Schost (2012): $\tilde{O}((\log q)^8)$;
- Genus 2 with RM: G.-Kohel-Smith (2011): $\tilde{O}((\log q)^5)$;
- Genus 3: ??? $\tilde{O}((\log q)^{14})$ mentioned here and there.

Second goal: give the exponent for genus 3 with and without RM.

Hyperelliptic curves

Def. A curve is hyperelliptic if it admits an equation

$$y^2 = f(x),$$

with f a monic, squarefree polynomial.

Remarks:

- In characteristic 2, need to modify the equation;
- We assume $\deg f$ is odd (imaginary model); enough for theoretical complexity (maybe not in practice). Then $\deg f = 2g + 1$ where g is called the genus;
- Have to think about the desingularized, projective model;
- There is only one point at infinity after desingularization: P_∞;
- The Jacobian is an associated Abelian variety of dimension g.

Divisors

Let $\mathrm{Div}_{\mathcal{C}}$ be the **free group** of points of $\mathcal{C}$:

$$\mathrm{Div}_{\mathcal{C}} = \Big\{ D = \sum_{P \in \mathcal{C}(\overline{\mathbb{F}_q})} n_P P \mid \text{for almost all } P,\ n_P = 0 \Big\}.$$

The **degree** of $D \in \mathrm{Div}_{\mathcal{C}}$ is $\deg D = \sum n_P$.

The divisor of a non-zero function $\varphi \in \overline{\mathbb{F}_q}(\mathcal{C})$ is

$$\mathrm{div}(\varphi) = \sum \mathrm{val}_P(\varphi) P,$$

where $\mathrm{val}_P(\varphi)$ is the valuation of φ at P.
The set of such divisors is the group of **principal divisors**:

$$\mathrm{Prin}_{\mathcal{C}} = \Big\{ \mathrm{div}(\varphi) \mid \varphi \in \overline{\mathbb{F}_q}(\mathcal{C})^* \Big\}.$$

Thm. A principal divisor has degree 0.

Divisor class group and Jacobian

Divisor class group:

$$\mathrm{Pic}^0_{\mathcal{C}} = \{\text{Degree-0 divisors}\}/\{\text{Principal divisors}\}.$$

This can be given the geometrical structure of a principally polarized **Abelian variety**: the **Jacobian** of $\mathcal{C}$, and we denote it $\mathrm{Jac}_{\mathcal{C}}$.

Rem. A purely geometric definition of $\mathrm{Jac}_{\mathcal{C}}$ can be done via an embedding in a projective space with theta functions.

Mumford representation

By Riemann-Roch theorem, each class has a unique representative of the form

$$D = P_1 + \cdots + P_r - r\,P_\infty, \text{with } r \leq g,$$

and no two P_i's are symmetric w.r.t the x-axis.

Thm. (**Mumford representation**) Any divisor class can be uniquely represented by a pair $\langle u(X), v(X) \rangle$, where

- u is monic, of degree at most g;
- $\deg v < \deg u$;
- u divides $v^2 - f$;

If D is as above, then $u(X) = \prod(X - x_{P_i})$ and $v(x_i) = y_i$.

Cantor's algorithm allows to compute efficiently in the Jacobian when elements are represented like this.

Weil's theorem

$$Z(\mathcal{C}, T) = \frac{P_{\mathcal{C}}(T)}{(1-T)(1-qT)},$$

Weil's theorem implies:

- $P_{\mathcal{C}}(T) = \prod_{i=1}^{2g}(1 - u_i T)$, where $|u_i| = q^{1/2}$;
- if $P_{\mathcal{C}}(T) = a_0 + a_1 T + \cdots a_{2g} T^{2g}$, then we have $a_{2g-i} = q^{g-i} a_i$;
- the coeffs are bounded by $\binom{2g}{g} q^g$ (could be more precise).

Link with the Frobenius endomorphism:

Let π be the $x \mapsto x^q$ map extended to a map from $\mathcal{C}$ to itself and then linearly to $\mathrm{Jac}_{\mathcal{C}}$ to itself. It can be proven that

$$\tilde{P}_{\mathcal{C}}(\pi) = 0,$$

where $\tilde{P}_{\mathcal{C}}$ is $P_{\mathcal{C}}$ with reversed-ordered coefficients.

We write $\chi_\pi(T) = \tilde{P}_{\mathcal{C}}(T)$ for this **characteristic polynomial of Frobenius**.

Plan

Introduction, background

Schoof's algorithm

Tools for polynomial systems

New complexity for large genus

The case of genus 3 with real multiplication

Torsion

Let A be an Abelian variety over $\mathbb{F}_q$ (A will be $\mathrm{Jac}_{\mathcal{C}}$).
The ℓ-**torsion subgroup** is

$$A[\ell] = \{P \in A(\overline{\mathbb{F}_q}) \mid \ell \cdot P = 0\}.$$

Thm. For a prime ℓ coprime to q, the **group structure** of $A[\ell]$ is

$$A[\ell] \cong (\mathbb{Z}/\ell\mathbb{Z})^{2g}.$$

The set $A[\ell] \setminus \{0\}$ is an algebraic variety of dimension 0, and we can consider its ideal.
Def. The **ideal** corresponding to the non-zero ℓ-torsion points is denoted by I_ℓ.

Rem. I_ℓ depends on the **set of coordinates** chosen to represent A. This could be projective coordinates, or a local affine patch.

Frobenius action on $A[\ell]$

Matrix representation of Frobenius.
The Frobenius endomorphism π maps elements of $A[\ell]$ to $A[\ell]$. Viewing $A[\ell]$ as an $\mathbb{F}_\ell$-vector space of dimension $2g$, π acts **linearly** on this vector space: it can be represented as a matrix, whose characteristic polynomial is $\chi_{\mathcal{C}}(\pi) \bmod \ell$.

Thm. The **characteristic polynomial** of π on $A[\ell]$ is the **reduction** mod ℓ of the **global** characteristic polynomial of π.

If I_ℓ is an ideal in a coordinate ring $\mathbb{F}_q[\overline{X}]$, the generic ℓ-torsion element is represented by the algebra $B_\ell = \mathbb{F}_q[\overline{X}]/I_\ell$.
Assuming computing in B_ℓ is efficient, we can compute $\chi_{\mathcal{C}}(\pi)$ mod ℓ.
Note: "efficient" is not so simple to define, here.

Combining modular information

Main point counting algorithm: (à la Schoof)

1. While the product of ℓ's already handled is $< \binom{2g}{g} q^g$:
   1.1 Pick the next small prime ℓ coprime to q;
   1.2 Compute the ℓ-torsion ideal I_ℓ;
   1.3 Find an efficient representation of I_ℓ;
   1.4 Compute $\chi_C(\pi) \mod \ell$;
2. Reconstruct $\chi_C(\pi)$ by CRT.

Rem. The number and the size of the ℓ's is **polynomial** in $g \log q$. But the ideal I_ℓ is of degree ℓ^{2g}, which is **exponential** in g.

Rem. The step 1.3 does not exist in the elliptic case, where we use the division polynomial ψ_ℓ to represent I_ℓ.
But 1.3 is the most important step for higher genus.

Coordinate systems for I_ℓ

An efficient representation starts with a coordinate system.

Theta functions:

- Need many coordinates, at least 2^g;
- But nice projective embedding: less non-genericity to handle.

Mumford coordinates:

- Optimal number of coordinates $O(g)$;
- But local affine coordinates: many non-generic cases if an intermediate point is not in this affine patch.

Plan

What do we want?

Coordinates of a generic ℓ-torsion element will be in

$$B_\ell = \mathbb{F}_q[\overline{X}]/I_\ell,$$

where $\overline{X}$ is the set of $2g$ Mumford coordinates.

Applying Frobenius = raising to the q-th power in B_ℓ.
This means being able to **work "modulo the ideal"**.
This is essentially the definition of a **Gröbner basis**.

Rem. We are interested both in proven complexity bounds and practical efficiency.

Gröbner bases – F4 / F5 algorithm

What is it?

- Algorithm that computes a Gröbner basis of the ideal, for any monomial order; (Faugère)
- Usually done in two steps: GB for `grevlex` and then change of ordering for `lex`;
- Heavily relies on linear algebra.

Good points, bad points.

✗ Bad complexity bounds if nothing is known.

✗ Good complexity bounds require hard-to-prove properties of the input system.

✗ Really compute the GB: need to take care about parasite components (saturation).

✓ Robust to many situations.

✓ Some public and efficient implementations.

Resultants (univariate)

What is it ?

- Algorithm to compute a combination of two input polynomials, with one less variable;
- Produces an element in the ideal: need to repeat to produce a generating set;
- Polynomial arithmetic;
- There exist multivariate resultants, but mostly of theoretical interest.

Good points, bad points.

- ✗ Not always easy to guarantee that we have a complete set of generators;
- ✗ Really bad complexities when there are many variables;
- ✓ Complexity bound do not assume too much on the input system;
- ✓ Some public and efficient implementations.

Geometric resolution

What is it ?

- Algorithm to put the system in triangular form, close to GB for `lex` order (Giusti, Lecerf, Salvy, Cafure, Matera, . . .);
- Incremental process based on Newton lifting;
- Relies on (univariate) polynomial arithmetic and (Jacobian) matrix inversion.

Good points, bad points.

- ✗ Intrinsically probabilistic (Monte Carlo);
- ✗ Only prototype implementations available;
- ✗ Requires some nice properties of the input system;
- ✓ Said properties easier to prove than for GB;
- ✓ Good complexity bounds.

XL

What is it ?

- Algorithm that compute a solution in a given field of definition (Courtois, Klimov, Patarin, Shamir, . . .)
- Same general idea as F4 (Lazard's algorithm using Macaulay matrices);
- Heavily relies on linear algebra.

Good points, bad points.

- ✗ Efficient only for solution with coordinates in a small finite field;
- ✗ Complexity bounds require hard-to-prove properties of the input system;
- ✓ Some public and efficient implementations (for basic XL);
- ✓ Sometimes heuristically more efficient than F4.

Summary of the situation for I_ℓ

The following is **specific to our case**.
Multi-homogeneity is an important property of our systems (see below).

	Applicable in theory	Applicable in practice	Can use multi-homog.
F4	?	✓	✓
Resultants	✓	✓	✗
Geom. resol.	✓	?	✓
XL	✗	✗	?

Rem. For your own problem, you'll have to write your own table.

Plan

Equations for the torsion (1)

Take a **generic divisor**:

$$D = \sum_{i=1}^{g} (P_i - P_\infty),$$

where $P_i = (x_i, y_i)$ and write $\ell D = 0$.

For any i, $\ell(P_i - P_\infty)$ is equivalent to a reduced divisor in Mumford representation:

$$\ell(P_i - P_\infty) = \langle u_i(X), v_i(X) \rangle,$$

where u_i and v_i are polynomials with coeffs that depends on x_i and y_i. They are exactly **Cantor's division polynomials**:

$$u_i(X) = \delta_\ell \left(\frac{x_i - X}{4y_i^2} \right), v_i(X) = \varepsilon_\ell \left(\frac{x_i - X}{4y_i^2} \right).$$

Equations for the torsion (2)

$$\ell D = 0 \iff \langle u_1(X), v_1(X)\rangle + \cdots + \langle u_g(X), v_1(X)\rangle = 0.$$

Applying $g-1$ times the group law: difficult to **control the degrees**.

Cantor sketched the following approach:
Consider the function

$$\varphi(X,Y) = P(X) + YQ(X)$$

and $\mathrm{div}\varphi = \langle u_1(X), v_1(X)\rangle + \cdots + \langle u_g(X), v_1(X)\rangle$.

Degrees of P and Q must be $\approx g^2/2$ (parity of $g-\ell$ plays a role).

Set g^2 **indeterminates** for the coefficients of P and Q. We have a system of equations

$$P(X) + \varepsilon_\ell\left(\frac{x_i - X}{4y_i^2}\right)Q(X) \equiv 0 \bmod \delta_\ell\left(\frac{x_i - X}{4y_i^2}\right).$$

Multi-homogeneity

This strategy **looks bogus**, because we have increased the number of variables from $O(g)$ to $O(g^2)$, and the degrees $O(\ell^2)$ of the equations did not decrease to compensate for it.

Def. A **multi-homogeneous** polynomial system is a set of equations $f_1(\overline{X}, \overline{Y}) = 0, \ldots, f_k(\overline{X}, \overline{Y}) = 0$, in two blocks of variables, where for each equation, the degree in $\overline{X}$ is $\leq d_X$ and the degree in $\overline{Y}$ is $\leq d_Y$.

Key quantity for complexity analysis:

$$d_x^{n_x}\, d_y^{n_y},$$

where n_x and n_y are the number of variables in each block.

We have added g^2 variables, but they occur in degree 1, so this won't hurt the multi-homogeneous complexity.

Geometric resolution and multi-homogeneity

With the geometric resolution algorithms in the end, the **complexity** of solving the system **should be polynomial** in

$$d_x^{n_x}\, d_y^{n_y} = O_g(\ell^{2g}).$$

But for that, we **need** the input system to be

- 0-dimensional (need to clean-up any higher dimensional parasite component);
- radical (no multiple roots);
- a regular sequence (each equation cuts cleanly the previous ones).

Rem. The first system you write to describe an algebraic situation is **never** like this.

Technicalities to get a proven complexity

0-dimensional: careful when writing equations; any denominator clearing must come with the appropriate saturation. Corresponding non-generic sub-cases must be handled independently with other polynomial systems.

radicality: comes from the fact that the multiplication by ℓ map can not involve multiplicities, but care must be taken to ensure that we did not introduce new multiplicities in our equation.

regular sequence: need to make a random (linear) change of coordinates and apply a positive characteristic, multi-homogeneous variant of Bertini's theorem.

degrees: Cantor's paper on division polynomials does not provide all the degree bounds we need.

Main result

Thm. There is a probabilistic algorithm that given a hyperelliptic curve of genus g over a finite field $\mathbb{F}_q$ computes its local Zeta function in expected time $O_g((\log q)^{O(g)})$.

(before, the best known complexity was with a quadratic exponent)

Rem. We do not claim more than a purely theoretical complexity result. Don't try to implement it following all the steps of the paper; several parts deal with things that should almost never occur in practice.

Plan

Introduction, background

Schoof's algorithm

Tools for polynomial systems

New complexity for large genus

The case of genus 3 with real multiplication

Equations for the torsion in genus 3

For genus 3, the equation for the torsion becomes $\ell D = 0 \Leftrightarrow$

$$\langle u_1(X), v_1(X)\rangle + \langle u_2(X), v_2(X)\rangle + \langle u_3(X), v_3(X)\rangle = 0,$$

$$\text{where} \quad u_i(X) = \delta_\ell\left(\frac{x_i - X}{4y_i^2}\right), v_i(X) = \varepsilon_\ell\left(\frac{x_i - X}{4y_i^2}\right).$$

Here, the indeterminates are (x_1, y_1), (x_2, y_2), (x_3, y_3).

We **apply the group law** once, between the first two divisors and get

$$\langle u_{12}(X), v_{12}(X)\rangle = -\langle u_3(X), v_3(X)\rangle.$$

Now, u_{12} and v_{12}'s coefficients depend on x_1, y_1, x_2, y_2, (and we use the symmetries).

Rem. Computing this input system can be done by working in the appropriate function field and takes no time compared to solving it.

Two ways of solving the polynomial system

In theory, with resultants:

- The number of variables is low (essentially 3, because the y_i do not count);
- The intermediate degrees do not grow too much compared to the degree of I_ℓ;
- Complexity ends-up being quasi-quadratic in $\deg I_\ell$, which is better than the other approaches.

In practice, with F4:

- The F4 algorithm behaves surprisingly well on these systems;
- Absolutely no hope to prove this;
- Many unexpected degree falls during the computation

Rem. Experiments with F4 done with `Magma` and `tinyGB`. For resultants, time estimates based on `FLINT` and `NTL`.

Results for genus 3 curves (without RM)

Complexity result:

Thm. Point counting for genus 3 hyperelliptic curves over a finite field $\mathbb{F}_q$ can be done in time $\tilde{O}((\log q)^{14})$.

Practical result: Experiments for a curve of genus 3, over $\mathbb{F}_p$, with a 64-bit prime p, and $\ell = 3$.
All things put together, we get a system with

- 5 variables;
- 5 equations of degrees 7, 53, 54, 55, 26.

The system can be solved (with F4, in Magma) in

- 14 days;
- 140 GB of RAM.

The next prime $\ell = 5$ is already out of reach !

Real multiplication (RM)

G.-Kohel-Smith (2011): In genus 2, the **complexity drops** from $\tilde{O}((\log q)^8)$ to $\tilde{O}((\log q)^5)$, if an explicit **real** endormorphism is known.

Let's follow this path in genus 3

RM curves considered by Tautz, Top, and Verberkmoes (1991):

$$\mathcal{C}_t : \quad y^2 = x^7 - 7x^5 + 14x^3 - 7x + t, \quad (t \neq \pm 2)$$

Explicit RM endomorphism on $\mathrm{Jac}_{\mathcal{C}_t}$ (Kohel, Smith 2006):

$$\eta_7(x, y) = \langle X^2 + 11\, xX/2 + x^2 - 16/9, y \rangle,$$

and we have

$$\eta_7^3 + \eta_7^2 - 2\eta_7 - 1 = 0,$$

so that $\mathbb{Z}[\eta_7] \cong \mathbb{Z}[2\cos(2\pi/7)] \subset \mathrm{End}(\mathrm{Jac}_{\mathcal{C}_t})$.

Explicit RM kernel

Let $\ell =$ be a split prime in $\mathbb{Z}[\eta_7]$, for instance

$$(13) = (2 - \eta_7 - 2\eta_7^2)\,(-2 + 2\eta_7 + \eta_7^2)\,(3 + \eta_7 - \eta_7^2).$$

Then the kernel $\mathrm{Jac}_{\mathcal{C}_t}[13]$ decomposes as a **direct sum of the kernels** of these 3 endomorphisms of degree ℓ^2.

The same strategy as before will work, in theory with resultants, and in practice with F4.

E.g. for $\ell = 13$, we have to solve three systems with

- 5 variables,
- 5 equations of degrees 7, 44, 45, 46, 52.

Each of them is smaller than what we had for $\ell = 3$.

Results for genus 3 with RM

Complexity result:

> **Thm.** Point counting for genus 3 hyperelliptic curves over a finite field $\mathbb{F}_q$ with an explicit real multiplication endomorphism can be done in time $\tilde{O}((\log q)^6)$.

Practical result: Experiments for $\mathcal{C}_t$, with $t = 42$ over $\mathbb{F}_p$, with $p = 2^{64} - 59$:

Modular information obtained:

mod ℓ^k	#var	degree of each eq.	time	memory
2	—	—	—	—
4 (inert^2)	6	$7, 7, 14, 15, 15, 10$	1 min	negl.
3 (inert)	5	$7, 53, 54, 55, 26$	14 days	140 GB
$13 = \mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3$	5	$7, 44, 45, 46, 52$	3×3 days	41 GB
$7 = \mathfrak{p}_1^3$	5	$7, 35, 36, 37, 36$	3.5h	6.6 GB
$29 = \mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3$	5	$7, 92, 93, 94, 100$	$>3\times2$ weeks	>0.8 TB

Practical results for genus 3 with RM (con't)

For $\ell = 29$, we failed to find the torsion (note that over a small finite field, the GB computation finished).
For $\ell = 7$, only partial information was obtained but not used.
But we got $\chi_{\mathcal{C}}(T)$ mod $3 \times 4 \times 13 = 156$.

Final parallel collision search:
We used the low-memory variant (G., Schost, 2004) of the algorithm by Matsuo, Chao and Tsujii (2002).
The complexity is $O(p^{3/4}/m^{3/2})$, where $m = 156$ is the known modular information.

Here: 190,000 3d pseudo-random walks of average length 32,000,000 led to a useful collision, in about 105 days (done in parallel in a few hours).

Conclusion

New complexity bounds:

- Arbitrary genus: $O_g((\log q)^{O(g)})$ (previous exponent was quadratic);
- Genus 3: $\tilde{O}((\log q)^{14})$ in general
- Genus 3: $\tilde{O}((\log q)^{6})$ with explicit RM
- See also recent result by Abelard, for arbitray genus with RM.

Take-home message about polynomial systems:

- No tool is perfect in all situations;
- Proving (good) complexity bounds can be really, really hard;
- Look for multi-homogeneity in your favorite systems.

Our genus 3 RM curve

The curve $\mathcal{C}_{42}$ of equation

$$y^2 = x^7 - 7x^5 + 14x^3 - 7x + 42$$

over $\mathbb{F}_p$ with $p = 2^{64} - 59$ has characteristic polynomial

$$\chi(T) = T^6 - \sigma_1 T^5 + \sigma_2 T^4 - \sigma_3 T^3 + p\sigma_2 T^2 - p^2\sigma_1 T + p^3,$$

with

$$\begin{aligned} \sigma_1 &= 986268198, \\ \sigma_2 &= 35389772484832465583, \\ \sigma_3 &= 10956052862104236818770212244. \end{aligned}$$