

Point counting in families of hyperelliptic curves

Hendrik Hubrechts
Katholieke Universiteit Leuven
Belgique

What this talk is all about

1. Short introduction to the reasons of point counting
2. Kedlaya's algorithm
3. Applying deformation to Kedlaya's approach
4. Some things about differential equations
5. The algorithm and some results

Why point counting?

Main 'economical' reason for point counting is cryptography (but there are other reasons...)

- Some one-way functions are based on the hardness of the discrete logarithm problem (DLP)
- Groups with hard DLP:
 - $\mathbb{F}_q^\times, \cdot$ (subexponential)
 - Jacobian of HEC/ \mathbb{F}_{p^n} (exponential if genus and (p or n) small)
- Subexponential algorithms:
 - \mathbb{F}_q^\times : index calculus
 - Jacobian (intermediate field): Claus Diem
 - Jacobian (higher genus): Adleman-DeMarrais-Huang e.a.

What are hyperelliptic curves?

A concrete ‘working definition’ of a hyperelliptic curve.

Let p be a prime, $n > 0$ an integer, \mathbb{F}_{p^n} a field of size p^n .

- p odd $\rightarrow \bar{\mathcal{C}} : Y^2 = Q(X)$, with $\deg Q = 2g + 1$
 Q has no double roots and is monic, g is the *genus*
- $p = 2 \rightarrow \bar{\mathcal{C}} : Y^2 + h(X)Y = f(X)$, with $\deg f = 2g + 1$,
 $\deg h \leq g$, no affine singularities

If the genus g equals 1, we have an *elliptic curve*. The *Jacobian* of \mathcal{C} is a finite ‘computable’ abelian variety connected with the curve over \mathbb{F}_{p^n} .

Lots of orders, ordered onorthodox

For 'assuring' the security of the DLP on the Jacobian, we have to know its order (it needs big prime factors)

Let $\bar{\mathcal{C}}$ be defined over \mathbb{F}_q , then (suppose $\bar{\mathcal{C}}$ to be *complete*)

$$Z(T) := \exp \left(\sum_{k=1}^{\infty} \frac{\#(\bar{\mathcal{C}}/\mathbb{F}_{q^k}) T^k}{k} \right) = \frac{f(T)}{(1-T)(1-qT)}$$

is the *zeta function* of the curve, a rational function with a lot of structure in it.

PROPERTY. $f(1)$ equals the number of points of the Jacobian over \mathbb{F}_q .

We will try to compute $Z(T)$ for *small* p !

Some algorithms out in the open

A short and highly inaccurate overview of some existing algorithms

- Naive, just count all points, exponentially
- Elliptic curves: Satoh, $\tilde{O}(n^3)$
- Elliptic curves: Harley, $\tilde{O}(n^2)$
- Characteristic 2: AGM (very low genus), $\tilde{O}(n^2)$
- Hyperelliptic, char odd: Kedlaya, $\tilde{O}(n^3)$ time and memory
- Hyperelliptic, char 2: Denef and Vercauteren, $\tilde{O}(n^3)$ time and memory

Monsky and Washnitzer's basic idea

We suppose we are given a hyperelliptic curve over \mathbb{F}_q , $q = p^n$:

$$Y^2 = Q(X).$$

Points over \mathbb{F}_q are precisely fixed points under the ('big') Frobenius map $x \mapsto x^q$ (over an algebraic closure).

Try to find some good cohomology theory with such an operator. Then the Lefschetz fixed point theorem might give you some result.

Monsky and Washnitzer came with the following theory.

p -Adic numbers

- $\mathbb{Q}_p := \{a = a_N p^N + a_{N+1} p^{N+1} + \dots, N \in \mathbb{Z}, a_i \in \{0, 1, \dots, p-1\}\}$
- If $a_N \neq 0$: order of a is N , ring of integers \mathbb{Z}_p : order ≥ 0
- \mathbb{Q}_q : unramified extension of degree n of \mathbb{Q}_p
- \mathbb{Z}_q gives residue field $\mathbb{Z}_q/p\mathbb{Z}_q \cong \mathbb{F}_q$
- $\sigma : \mathbb{Q}_q \rightarrow \mathbb{Q}_q :=$ lift of Frobenius automorphism $\mathbb{F}_q \mapsto \mathbb{F}_q : x \mapsto x^p$

By ‘lifting a polynomial $Q(X)$ to characteristic zero’ we mean:
lift all nonzero coefficients to $\mathbb{Z}_q \setminus p\mathbb{Z}_q$ and keep the zeroes.

Some more technical background

Weakly convergent, or overconvergent, power series:

$$\mathbb{Q}_q\langle X \rangle^\dagger := \left\{ \sum_{i=0}^{\infty} a_i X^i \text{ s.t. } \exists \alpha > 0, \beta : \text{ord } a_i \geq \alpha \cdot i + \beta \right\}$$

Analytic meaning: these functions converge on some disk strictly larger than the unit disk.

The MW construction

Lift the curve \bar{C} to characteristic zero: $C : Y^2 = Q(X)$, the coordinate ring is then $A = \frac{\mathbb{Z}_q[X, Y]}{Y^2 - Q(X)}$. Using weak completions

$$A^\dagger := \frac{\mathbb{Z}_q\langle X, Y \rangle^\dagger}{Y^2 - Q(X)}$$

and defining a differential $d : A^\dagger \rightarrow D^1(A^\dagger)$ we obtain

$$H_{\text{MW}}^1(\bar{C}) := \frac{D^1(A^\dagger)}{dA^\dagger} \otimes \mathbb{Q}_q,$$

the first Monsky Washnitzer cohomology space. $H_{\text{MW}}^1(\bar{C})$ does not depend on the chosen lift and has dimension $2g$ over \mathbb{Q}_q .

A lift of Frobenius does exist on $H_{\text{MW}}^1(\bar{C})$.

How Kedlaya made it practical

Kedlaya did two essential things to get a decent algorithm.

1. Removing the ramification points allowed an easy Frobenius.

Let \tilde{C} be the new curve.

2. An isomorphism

$$H_{\text{DR}}^1(\tilde{C}) \rightarrow H_{\text{MW}}^1(\tilde{C})$$

exists canonically, hence computing is possible.

- surjectivity: ‘Kedlaya’s lemma’
- isomorphism: vector space dimensions

More concrete

Removing the ramification points \rightarrow

$$A^\dagger = \frac{\mathbb{Z}_q \langle X, Y, Y^{-1} \rangle^\dagger}{Y^2 - Q(X)},$$

with corresponding $H_{\text{MW}}^1(\tilde{\mathcal{C}})$. Frobenius acts on it as $X \mapsto X^p$,
 $dX \mapsto pX^{p-1}dX$ and

$$Y \mapsto Y^p \left(1 + \frac{Q^\sigma(X^p) - Q(X)^p}{Q(X)^p} \right)^{1/2}.$$

Kedlaya shows it is sufficient to consider $H_{\text{MW}}^{1-}(\tilde{\mathcal{C}})$, a $2g$ -dimensional subspace with basis

$$\left\{ \frac{X^i}{Y^3} dX \right\}_{i=0}^{2g-1}.$$

Finishing Frobenius

We now have an operator on a $2g$ -dimensional vector space, hence a matrix F over \mathbb{Q}_q . The ‘big’ Frobenius will have as matrix

$$\mathcal{F} = \prod_{i=0}^{n-1} F^{\sigma^i}.$$

Lefschetz fixed point formula implies

- nominator of $Z(T)$: $f(T) = \det(1 - \mathcal{F}T)$
- size of Jacobian equals $f(1) = \det(1 - \mathcal{F})$

In practice: Compute Frobenius on the basis elements and reduce using certain easy reduction formulae, compute F from it, then \mathcal{F} , and finally take the characteristic polynomial.

Deform until it's easy

We embed the curve $\bar{\mathcal{C}} : Y^2 = Q(X)$ in a family

$$\bar{\mathcal{C}}_{\Gamma} : Y^2 = Q(X, \Gamma)$$

s.t. for some concrete $\bar{\gamma}$ we have $\bar{\mathcal{C}}_{\bar{\gamma}} = \bar{\mathcal{C}}$.

Imitating Kedlaya's construction gives a matrix $F(\Gamma)$ s.t. for a good lift γ of $\bar{\gamma}$ we have $F(\gamma) = F_{\bar{\mathcal{C}}}$.

In *certain cases* we gain

- memory will be $\mathcal{O}(n^2)$
- time can be made $\mathcal{O}(n^{2.667})$
- (but not together, unfortunately)

Note. Some kind of deformation was first developed by Dwork, and exploited in some situations by Lauder and Tsuzuki.

What are 'certain cases'?

Choose $Y^2 = Q(X, \Gamma)$ over a **small field** \mathbb{F}_q , monic in X , s.t. $Y^2 = Q(X, \bar{\gamma})$ gives the desired curve, $\bar{\gamma} \in \mathbb{F}_{q^n}$.

Some $\bar{\gamma}$ do not give a HEC, namely the zeroes of

$$\bar{r}(\Gamma) := \text{Res}_X \left(Q(X, \Gamma), \frac{\partial}{\partial X} Q(X, \Gamma) \right).$$

We suppose that $\Gamma = 0$ is ok (can be achieved with a translation).

Not all curves over \mathbb{F}_{q^n} can be constructed this way!

Decreasing/increasing the characteristic

Lift $Q(X, \Gamma)$ to characteristic zero as explained before, and define again the resultant

$$r(\Gamma) := \text{Res}_X \left(Q(X, \Gamma), \frac{\partial}{\partial X} Q(X, \Gamma) \right).$$

A *Teichmüller lift* of $\bar{\gamma}$ in \mathbb{F}_{q^n} is the unique lift γ of $\bar{\gamma}$ such that $\gamma^{q^n} = \gamma$.

The set of ‘good lifts’ consists of the Teichmüller lifts of non-zeroes of r modulo p . **Remark:** we require that the resultant is ‘monic’!

Complicating \mathbb{Q}_q and A^\dagger

\mathbb{Q}_q -vector spaces become S -modules where

$$S := \mathbb{Q}_q \langle \Gamma, r(\Gamma)^{-1} \rangle^\dagger.$$

(Reason: $r(\Gamma)^{-1}$ comes from the reduction formulae.)

The S -module that represents $A^\dagger \otimes \mathbb{Q}_q$:

$$T := \frac{\mathbb{Q}_q \langle X, Y, Y^{-1}, \Gamma, r(\Gamma)^{-1} \rangle^\dagger}{Y^2 - Q(X, \Gamma)}.$$

Descending into cohomology

As before: $d : T \mapsto TdX$, $d = \frac{\partial}{\partial X} dX$, and

$$\frac{TdX}{dT} \cong H^- \oplus H^+,$$

where H^- is a free $2g$ -dimensional S -module with basis as before.

We can define Frobenius on H^- by extending it with $\Gamma \mapsto \Gamma^p$ and $d\Gamma \mapsto p\Gamma^{p-1}d\Gamma$.

Result: $F(\Gamma)$ as matrix over S s.t.

$F(\gamma)$ is what we need for a Teichmüller lift γ .

This method is however inefficient, working similar to Kedlaya's algorithm would again require memory $\mathcal{O}(n^3)$.

Connecting two worlds: the connection

On T we can define a *connection*

$$\nabla : T \rightarrow Td\Gamma : t(X, \Gamma) \mapsto \frac{\partial}{\partial \Gamma} t(X, \Gamma) d\Gamma,$$

which extends to

$$\nabla : \frac{TdX}{dT} \rightarrow \frac{TdXd\Gamma}{d(Td\Gamma)}.$$

So ∇ defines an operator on H^- , say with matrix G over S .

Finally that differential equation

Summarizing the above maps we find

$$\begin{array}{ccc} H^- & \xrightarrow{\nabla} & H^- d\Gamma \\ \downarrow \text{Frob} & & \downarrow \text{Frob} \\ H^- & \xrightarrow{\nabla} & H^- d\Gamma, \end{array}$$

a *commutative* diagram.

This implies the differential equation ($\dot{\cdot}$ stands for $\partial/\partial\Gamma$)

$$\dot{F} + F \cdot G = p\Gamma^{p-1} G^\sigma(\Gamma^p) \cdot F.$$

Don't forget the even primes!

For $p = 2$ everything is more complicated, but in the end we find:

- a similar S -module of dimension $2g$
- with a Frobenius that satisfies the same equation

$$\dot{F} + F \cdot G = 2\Gamma G^\sigma(\Gamma^2) \cdot F.$$

How can we work concretely with infinite objects?

Due to the Weil conjectures: sufficient to compute \mathcal{F} modulo a certain power of p , which is $\mathcal{O}(ng)$.

We work always ‘absolutely’ modulo p^N for some well chosen N , and with power series modulo Γ^M . Both $N, M = \tilde{\mathcal{O}}(n)$.

Three problems we need to address

1. We need elements from S , but get power series. . .
2. How can one solve such a differential equation?
3. Dividing by p decreases the accuracy, how to cope with this?

Solving the differential equation

$$\dot{F} + FG = p\Gamma^{p-1}G^\sigma(\Gamma^p)F$$

Towards finite matrices

The matrix G of ∇ as power series has size $N \cdot M$ (or ∞).

We can make it smaller

- Odd p : $H(\Gamma) := r(\Gamma) \cdot G(\Gamma)$ gives polynomials of low degree
- $p = 2$: exist small matrices B and D such that

$$D = \dot{B} + BG,$$

with $\det B \approx r(\Gamma)$, i.e. $\det B$ is invertible in S .

Towards a convergent solution

The entries of $F(\Gamma)$ are overconvergent as power series

$$\sum_{i \in \mathbb{Z}} \frac{a_i(\Gamma)}{r(\Gamma)^i},$$

i.e. $\text{ord}(a_i) \geq \alpha \cdot |i| + \beta$.

If we know bounds for α and β , and we want our result modulo p^k , then it's easy to find χ s.t.

multiplying with r^χ gives a polynomial mod p^k .

Conclusion: we need $r^\chi F$ for some computable χ .

Cheap solution: Let $\tilde{F} := r^\chi F$, plug this in the differential equation:

$$r\dot{\tilde{F}} + r\tilde{F}G = \chi\dot{r}\tilde{F} + p\Gamma^{p-1}rG^\sigma\tilde{F}.$$

Why not being more general?

All cases, including $p = 2$, can be covered by one single equation

$$A\dot{F}B + XFB + AFY = 0$$

1. A , B , X , Y are defined over $\mathbb{Q}_q[\Gamma]$ and of 'low degree'
2. $A_0 = A(\Gamma = 0)$ and $B_0 = B(\Gamma = 0)$ are invertible
3. Boundary condition $F(\Gamma = 0) = F_0$ is given

Two methods to solve this

$$A\dot{F}B + XFB + AFY = 0$$

- (a) Solve in parts: $A\dot{F}_1 + XF_1 = 0$ and $\dot{F}_2B + F_2Y = 0$,
with $F_1(0) = F_0$ and $F_2(0) = 1$, then $F = F_1F_2$
- (b) Solve the equation directly

Theoretically (a) is slightly faster, using fast polynomial multiplication.
In practice (b) is a lot faster.

A timing example, genus 2, 3^{200}

a				b
F_1	F_2	$F = F_1F_2$	$r^\times F$	$r^\times F$
42	14	1565	424	303

Short overview of the algorithm

1. Lift $Y^2 = Q(X, \Gamma)$ to \mathbb{Q}_q
2. Compute matrix F_0 of the small Frobenius at $\Gamma = 0$ (à la Kedlaya)
3. Compute matrices H (char odd), B and D (char 2) using reduction formulae similar to Kedlaya's
4. Determine $r(\Gamma)^\times F(\Gamma)$ from the corresponding differential equation
5. Calculate the Teichmüller lift $\gamma \in \mathbb{Q}_{q^n}$ of $\bar{\gamma} \in \mathbb{F}_{q^n}$
6. Substitute γ : $F := r(\gamma)^\times F(\gamma)$
7. Compute $\mathcal{F} := \prod_{i=0}^{\log_p(q^n)-1} F^{\sigma^i}$
8. Output the zeta function $\det(1 - \mathcal{F}T) / ((1 - T)(1 - qT))$

Short overview of the complexity

1. Up to computing $r(\Gamma) \times F(\Gamma)$: $\tilde{\mathcal{O}}(n^2)$
2. Every next step requires $\tilde{\mathcal{O}}(n^3)$
3. Biggest object in memory is $r(\Gamma) \times F(\Gamma)$: $\mathcal{O}(n^2)$
4. With fast arithmetic (see further): up to $F = F(\gamma)$: $\tilde{\mathcal{O}}(n^2)$
5. Central problem: compute $\sigma^k(x)$ for $x \in \mathbb{Q}_{p^n}$ for $k = \mathcal{O}(n)$
6. By fast modular composition of polynomials:
 $\mathcal{O}(n^{2.667})$ with space $\mathcal{O}(n^{2.5})$
7. By fast multipoint evaluation: $\mathcal{O}(n)$ curves in time $\tilde{\mathcal{O}}(n^3)$
8. **Someone suggestions for $\sigma^k(x)$?**

Let the computer do the work

System: Magma 2.12-14 on a Pentium IV running SuSE at 2.4 GHz.

First column: precomputation/‘one parameter’, second: memory

$p^n \backslash g$	2		3		4	
3^{100}	125/305	24.44	1942/1818	98	11626/5110	352
(K) 3^{100}	1216	110.10	6104	399	17483	723
3^{200}	307/2163	41.92	4403/12130	197		
(K) 3^{200}	11785	635.34				
5^{100}	397/632	34.58	4984/2592	152	27654/9367	546
(K) 5^{100}	2796	221.30	16082	606	46842	1280
31^{100}	7178/4303	217.33				

$g = 1$	3^{100}		3^{500}		7^{500}		13^{500}	
	2.4/34.5	11.5	14.6/4377	45.5	73.4/10389	74.5	221/13299	89

Using fast p -adic arithmetic

Combining a suggestion of Lauder with an algorithm of Harley

- A **Teichmüller modulus** for \mathbb{Q}_{p^n} is some $\varphi(x)$ s.t.

$$\mathbb{Q}_{p^n} = \frac{\mathbb{Q}_p[x]}{\varphi(x)} \text{ and } x^{p^n} = x, \bar{x} = \bar{\gamma}.$$

Equivalently: $\bar{\varphi}(x)$ is the minimal polynomial of $\bar{\gamma}$ and $\varphi(x) | x^{p^n} - x$.

- Harley gave an amazingly fast algorithm to compute φ (for very small p)
- Evaluating $f(\Gamma)$ in γ is then just $f(x) \bmod \varphi(x)$

Even faster — examples

field	3^{1000}		5^{1000}		7^{1000}		2^{2000}		2^{10000}	
mod p^N	200		200		200		400		2000	
degree f	5000		5000		5000		10000		50000	
MinPol	0	2.13	0	3.06	0	4.18	0	1.01	0	78.27
φ	0	4.48	0	74.21	0	1588	0	1.77	0	123
Teichmüller	1214	0	1952	0	3856	0	4964	0	-	0
Frobenius	351	0.47	425	1.28	511	1.88	877	0.47	-	31
Evaluation	395	0.91	418	1.02	675	0.6	1098	1.07	-	122

Some final results with this fast arithmetic

Built-in arithmetic in Magma versus fast arithmetic à la Harley

- $g = 1, 3^{500}$ 4391 (45MB) / 172 (28MB)
- $g = 1, 3^{1000}$ 36709 (236MB) / 1269 (75MB)
- $g = 1, 3^{4000}$ - (-) / 124049 (954MB) (34.5 hours)
- $g = 2, 3^{200}$ 2472 (42MB) / 450 (44MB)
- $g = 3, 3^{200}$ 16540 (197MB) / 5413 (222MB)
- $g = 4, 3^{100}$ 16745 (352MB) / 14043 (405MB)

— — — —