

Computing zeta functions of projective hypersurfaces in large characteristic

David Harvey

New York University

22nd April 2010

Workshop on Counting Points: Theory, Algorithms and Practice
Centre de recherches mathématiques (CRM), Montréal, Canada

$$q = p^a$$

X = variety over \mathbf{F}_q

$$Z_X(T) = \exp \left(\sum_{r \geq 1} \frac{\#X(\mathbf{F}_{q^r})}{r} T^r \right) \in \mathbf{Q}(T)$$

Goal: compute $Z_X(T)$ efficiently when p is “large”

(Time = bit-complexity)

Schoof (1985) and descendants, ℓ -adic/CRT method:

Curve of genus g in time

$$(a \log p)^{g^{O(1)}}.$$

Asymptotically best known approach for fixed g and large p .

Not available for higher-dimensional varieties (except abelian varieties).

Lauder (2004), p -adic deformation method:

Degree d smooth hypersurface $X \subset \mathbf{P}^n$ in time

$$p^{2+\epsilon} \text{poly}(d^n a).$$

Dense input size is $d^n a \log p$ bits.

Note $p^{2+\epsilon}$ contribution is *independent of dimension*.

H. (2007), based on Kedlaya's p -adic algorithm:

Genus g hyperelliptic curve in time

$$p^{0.5+\epsilon} \text{poly}(ga).$$

e.g. $g = 3$, $q \approx 3 \times 10^{16}$ is feasible (30 hours on single CPU)

Minzlaff (2008): superelliptic curve in time

$$p^{0.5+\epsilon} \text{poly}(ga).$$

Main question

Can we obtain $p^{0.5+\epsilon}$ for curves more general than superelliptic?
For varieties of higher dimension?

Theorem (tentative)

Let $X \subset \mathbf{P}_{\mathbf{F}_q}^n$ be a smooth projective hypersurface of degree d ($p \nmid d$), satisfying a nondegeneracy condition (see next slide).

Then $Z_X(T)$ can be computed in time

$$p^{0.5+\epsilon} d^{n^2+O(n)} a^{n+O(1)},$$

and $\#X(\mathbf{F}_q)$ can be computed in time

$$p^{0.5+\epsilon} d^{O(n)} a^{n+O(1)}.$$



Back-of-the-envelope running time estimates...
Still many details to be checked...

Nondegeneracy condition

Let $f \in \mathbf{F}_q[x_0, \dots, x_n]$ be homogeneous defining polynomial.

For $S \subseteq \{0, \dots, n\}$, let

$$J_S = \langle x_i \partial_i f \rangle_{i \in S} + \langle \partial_i f \rangle_{i \notin S}.$$

(Here $\partial_i = \partial / \partial x_i$.)

Nondegeneracy condition:

There exists S with $d + |S| \geq n + 1$ such that J_S defines the empty scheme, i.e. $\text{rad } J_S = (x_0, \dots, x_n)$.

Geometric interpretation: for all subsets $T \subseteq S$, the intersection of X with the coordinate hyperplanes defined by $\{x_i\}_{i \in T}$ is smooth.

If $d > n$, can take $S = \emptyset$, equivalent to X itself being smooth.

Implementation

A toy implementation in Sage has existed for a few weeks.

Ran on a random degree 4 hypersurface in \mathbf{P}^3 :

$$Z_X(T) = ((1 - T)(1 - qT)(1 - q^2T)P(T))^{-1}$$

where $\deg P = 21$.

Precision parameters selected experimentally to determine $P(T)$ unambiguously (not proved correct).

- ▶ \mathbf{F}_{11} : 29 hours. Checked $\#X(\mathbf{F}_{11^r})$ for $1 \leq r \leq 4$ against more-or-less naive point count in Magma.
- ▶ \mathbf{F}_{101} : 80 hours. Checked for $1 \leq r \leq 2$.
- ▶ \mathbf{F}_{1009} : 239 hours. Checked for $r = 1$.
- ▶ \mathbf{F}_{10007} : 963 hours (+ segfault on exit). Checked for $r = 1$.

AKR algorithm

Algorithm based on AKR = Abbott–Kedlaya–Roe (“Bounding Picard numbers of surfaces using p -adic cohomology”, 2005).

$X = \{f = 0\} \subset \mathbf{P}_{\mathbf{F}_q}^n$ defined by $f \in \mathbf{F}_q[x_0, \dots, x_n]$, $\deg f = d$

$U = \mathbf{P}^n \setminus X$

$\sigma_q = q$ -th power Frobenius

$P(T) = \det(1 - q^{-1}\sigma_q T | H_{\text{rig}}^n(U))$

Then

$$Z_X(T) = P(T)^{(-1)^n} \prod_{i=0}^{n-1} \frac{1}{1 - q^i T}.$$

So it suffices to compute $P(T) \in \mathbf{Z}[T]$.

AKR algorithm

Plan: compute Frobenius action on $H_{\text{rig}}^n(U)$.

\tilde{f} = lift of f to $\mathbf{Z}_q[x_0, \dots, x_n]$

\tilde{U} = lift of U defined by \tilde{f} , i.e. with coordinate ring

$$\tilde{A} = \text{degree 0 piece of } \mathbf{Z}_q[x_0, \dots, x_n, z]$$

where $z = \tilde{f}^{-1}$, $\deg z = -d$.

Then

$$H_{\text{rig}}^n(U) \cong H_{\text{dR}}^n(\tilde{U}/\mathbf{Q}_q).$$

Explicit description of $H_{\text{dR}}^n(\tilde{U}/\mathbf{Q}_q)$ (Griffiths): let

$$\Omega = \sum_{i=0}^n (-1)^i x_i dx_0 \wedge \cdots (\text{omit } dx_i) \cdots \wedge dx_n.$$

Then $H_{\text{dR}}^n(\tilde{U}/\mathbf{Q}_q)$ is the quotient of

$$\langle z^m G \Omega : m \geq 1, \deg G = md - n - 1 \rangle$$

by

$$\langle (z^m \partial_i G - m z^{m+1} G \partial_i \tilde{f}) \Omega : 0 \leq i \leq n, m \geq 1, \deg G = md - n \rangle,$$

i.e. relations declare that exact forms are zero in cohomology.

AKR algorithm

A theorem of Macaulay implies that

$$(x_0, \dots, x_n)^\alpha \subset J$$

where $\alpha = (n+1)(d-2) + 1$. (Here $J = J_\emptyset = \langle \partial_i \tilde{f} \rangle$.)

This yields a *reduction algorithm* for computing in H_{dR}^n . Take any form $z^{m+1}F\Omega$ with $\deg F \geq \alpha$. Then $F = \sum_i G_i(\partial_i \tilde{f})$ for some polynomials G_i , and cohomology relations yield

$$z^{m+1}F\Omega \sim \frac{1}{m} z^m \sum_i (\partial_i G_i)\Omega.$$

Cannot reduce if $\deg F < \alpha$, but “left-over” monomials form a basis for H_{dR}^n (can be computed explicitly by easy linear algebra).

AKR algorithm

What about Frobenius?

Let $A^\dagger =$ weak completion of \tilde{A} (à la Monsky–Washnitzer):
elements are power series $\sum_{j \geq 0} G_j z^j$, where $G_j \in \mathbf{Q}_q[x_0, \dots, x_n]$,
 $\deg G_j = jd$, satisfying overconvergence condition
 $\liminf_{j \rightarrow \infty} v_p(G_j)/j > 0$.

Lift (absolute) Frobenius to A^\dagger via $x_i^\sigma = x_i^p$ ($0 \leq i \leq n$) and

$$\begin{aligned} z^\sigma &= \tilde{f}^{-\sigma} = (\tilde{f}^p - (\tilde{f}^p - \tilde{f}^\sigma))^{-1} \\ &= z^p (1 - z^p (\tilde{f}^p - \tilde{f}^\sigma))^{-1} \\ &= z^p \sum_{j \geq 0} z^{pj} (\tilde{f}^p - \tilde{f}^\sigma)^j. \end{aligned}$$

(Converges in A^\dagger since $p \mid \tilde{f}^p - \tilde{f}^\sigma$.)

Also

$$(z^t)^\sigma = z^{pt} \sum_{j \geq 0} \binom{t+j-1}{t-1} z^{pj} (\tilde{f}^p - \tilde{f}^\sigma)^j$$

and

$$\Omega^\sigma = p^n (x_0 \cdots x_n)^{p-1} \Omega.$$

Therefore we obtain series expansion for σ applied to a cohomology basis element:

$$(z^t x_0^{k_0} \cdots x_n^{k_n} \Omega)^\sigma.$$

Summary of AKR algorithm:

1. Compute a basis for H_{dR}^n
2. Compute series approximations for $(z^t x_0^{k_0} \cdots x_n^{k_n} \Omega)^\sigma$ for each basis element, to some p -adic and z -adic precision
3. Apply reduction algorithm to reduce each series back to basis elements
4. This yields matrix of absolute Frobenius; take product of conjugates to obtain matrix of q -th Frobenius
5. Characteristic polynomial is $P(T)$ to some p -adic precision

Choosing high enough precision determines $P(T)$ uniquely.

AKR did not analyse complexity.

Running time behaves at least like p^n , because algorithm works with dense polynomials like \tilde{f}^P in n variables.

New algorithm makes three modifications to AKR:

1. Use “sparse” series expansion for Frobenius
2. Use “controlled reduction”
3. Use “accelerated reduction”

(1) + (3) are similar to the approach in H. (2007) for hyperelliptic curves.

(2) seems to be new.

Sparse series expansion

First modification: use “sparse” series approximation.

$$\begin{aligned}(z^t)^\sigma &= z^{pt} \sum_{j \geq 0} \binom{t+j-1}{t-1} z^{pj} (\tilde{f}^p - \tilde{f}^\sigma)^j \quad (\text{same series as before}) \\ &\equiv z^{pt} \sum_{j=0}^{N-1} \binom{t+j-1}{t-1} z^{pj} (\tilde{f}^p - \tilde{f}^\sigma)^j \quad (\text{mod } p^N) \\ &= z^{pt} \sum_{j=0}^{N-1} \binom{t+j-1}{t-1} (1 - z^p \tilde{f}^\sigma)^j \\ &= \sum_{k=0}^{N-1} C_{t,k} z^{p(k+t)} \tilde{f}^{k\sigma} \quad \left[C_{t,k} = (-1)^k \sum_{j=k}^{N-1} \binom{t+j-1}{t-1} \binom{j}{k} \right]\end{aligned}$$

Number of terms does not depend on p !

Sparse series expansion

Thus $(z^t x_0^{k_0} \cdots x_n^{k_n} \Omega)^\sigma$ is approximated by a sum of terms of the form

$$z^{pm} x_0^{pr_0-1} \cdots x_n^{pr_n-1} \Omega,$$

i.e. on a lattice with distance p between terms.

Number of terms is about $O((Nd)^n)$.

To recover zeta function, need $N = O(d^n a)$.

To recover just $\#X(\mathbf{F}_q)$, need $N = O(na)$.

Sparse series expansion

Sparse series expansion is not enough...

Careless application of reduction algorithm leads to dense series with at least p^n terms!

Second modification: use “controlled reduction”.

Assume we have $S \subseteq \{0, \dots, n\}$ satisfying nondegeneracy condition.

Macaulay's theorem implies that

$$(x_0, \dots, x_n)^\alpha \subset J_S$$

where $\alpha = (n + 1)(d - 2) + |S| + 1$.

Controlled reduction

Take any differential $z^{m+1}x^u F\Omega$, where $\deg F = \alpha$, so $F \in J_S$.

Here $x^u = x_0^{u_0} \cdots x_n^{u_n}$. Think of $\deg x^u$ as “large”.

Write $F = \sum G_i(\partial_i \tilde{f})$, with $x_i \mid G_i$ for $i \in S$. Then

$$\begin{aligned} z^{m+1}x^u F\Omega &\sim \frac{z^m}{m} \sum_i \partial_i(x^u G_i)\Omega \\ &= \frac{z^m}{m} \frac{x^u}{\prod_{i \notin S} x_i} H\Omega \\ &= \frac{z^m}{m} \frac{x^u}{x_0^\ell \prod_{i \notin S} x_i} (x_0^\ell H)\Omega = \frac{z^m}{m} x^{u'} F'\Omega \end{aligned}$$

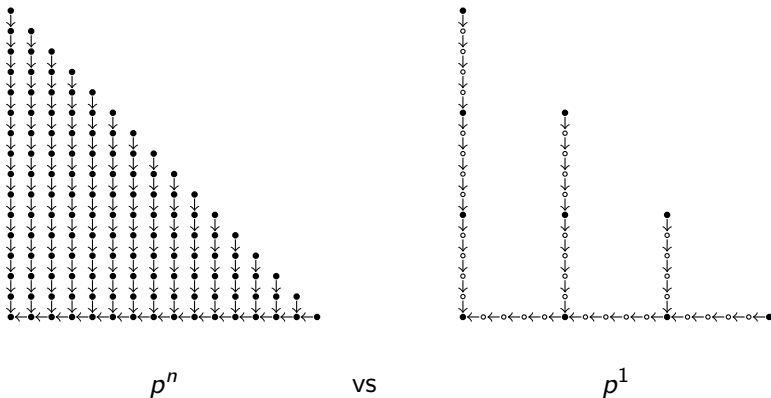
where $\ell = d - n - 1 + |S| \geq 0$ (by nondegeneracy condition).

In particular, $\deg F' = \alpha$. Now iterate!

Controlled reduction

Combined with sparse series expansion, already this technique reduces complexity from p^n to p^1 !

Schematically:



Accelerated reduction

Third modification: use “accelerated reduction”.

Consider the reduction described above, i.e.

$$z^{m+1}x^u F\Omega \implies \frac{z^m}{m}x^{u'} F'\Omega.$$

Let \bar{F} and \bar{F}' denote the vectors of coefficients. Then

$$\bar{F}' = S(m)\bar{F}$$

where $S(m)$ is a matrix of *linear polynomials* in m over \mathbf{Z}_q .

To iterate the reduction, want to compute the matrix product

$$S(m-p+1) \cdots S(m-1)S(m).$$

Accelerated reduction

Fortunately there is a fast algorithm for computing such products!

Let $S(m)$ be an $r \times r$ matrix of linear polynomials over a ring R .

Computing $S(k) \cdots S(1)S(0)$ naively requires

$$O(kr^\omega)$$

ring operations ($\omega \leq 3$ is exponent of matrix multiplication).

Better algorithm of Chudnovsky–Chudnovsky (1988), using fast polynomial arithmetic, obtains

$$O(k^{0.5+\epsilon}r^\omega)$$

ring operations.

Accelerated reduction

Sketch of Chudnovskys' algorithm for $S(m) = (m)$ ($r = 1$).

Want to compute $1 \times 2 \times \dots \times k$. For simplicity take $k = s^2$.

1. Compute $f(x) = (x + 1)(x + 2) \dots (x + s)$.
2. Evaluate simultaneously $f(0)$, $f(s)$, $f(2s)$, ..., $f((s - 1)s)$.
3. Multiply values together.

Cost of (1) and (2) is $O(s^{1+\epsilon}) = O(k^{1/2+\epsilon})$ using product tree & fast polynomial arithmetic.

See also improvements by Bostan–Gaudry–Schost (2004).

Summary of new algorithm:

1. Compute a basis for H_{dR}^n
2. Compute *sparse* series approximations for $(z^t x_0^{k_0} \cdots x_n^{k_n} \Omega)^\sigma$ for each basis element
3. Compute reduction matrices moving between adjacent lattice points
4. Apply reduction matrices to terms from step 2 to compute absolute Frobenius matrix; get $P(T)$ as before.

For large p , all the work is in step 3.

Open questions

1. Can the algorithm be made more practical?
 - ▶ countless opportunities for optimisation
 - ▶ implementation grunt work
2. Can $(d^n a)^{n+O(1)}$ be reduced to $(d^n a)^{O(1)}$?
 - ▶ maybe combine with deformation techniques?
(can we deform in $p^{0.5}$ time?)
3. Can we drop the (annoying) nondegeneracy condition?
4. What about smooth affine varieties?
5. Can we drop smoothness condition?
6. Can we do better than $p^{0.5}$?
 - ▶ does anyone know how to compute $(p-1)! \bmod p^2$ faster than $O(p^{0.5+\epsilon})$?

Thank you!