Controlled reduction in the p-adic cohomology of toric hypersurfaces

Kiran S. Kedlaya (joint work with David Harvey, UNSW)

Department of Mathematics, Massachusetts Institute of Technology; kedlaya@mit.edu
Department of Mathematics, University of California, San Diego; kedlaya@ucsd.edu

Number theory, algebraic geometry, and model theory
in honor of Jan Denef's 60th birthday
CIRM, Luminy, September 12, 2011

For slides, see http://math.mit.edu/~kedlaya/papers/talks.shtml.

Supported by NSF, DARPA, MIT, UCSD.

Contents

Contents

Zeta functions

Let X be an algebraic variety over a finite field $\mathbb{F}_q$. Let X° be the set of closed points of X. The *zeta function* of X is the power series

$$\zeta_X(T) = \prod_{x \in X^\circ} (1 - T^{\deg(x/\mathbb{F}_q)})^{-1}.$$

Many of its properties (e.g., the Weil conjectures) can be established using étale cohomology with coefficients in $\mathbb{Q}_\ell$, for ℓ any prime other than the characteristic of $\mathbb{F}_q$.

However, the properties of $\zeta_X(T)$ can also be obtained using p-adic analytic techniques, where p is the characteristic of $\mathbb{F}_q$. For instance, Dwork (1960) proved that $\zeta_X(T)$ represents a rational function in T; this predates the definition of étale cohomology!

Machine computation of zeta functions: motivation

Since the late 1990s, there has been a lot of work on algorithms to compute $\zeta_X(T)$ (and related objects) for various classes of algebraic varieties.

One original motivation came from cryptography, where it became necessary to compute orders of groups of points on elliptic curves over extremely large finite fields (e.g., $\mathbb{F}_{2^{256}}$). Subsequently, Jacobians of genus 2 curves were also needed.

However, there are plenty of mathematical reasons to be interested in such algorithms. One example from my work: investigating analogues of the Sato-Tate conjecture for genus 2 curves.

Nowadays, there is even some motivation from mathematical physics: arithmetic analogues of mirror symmetry.

Machine computation of zeta functions: motivation

Since the late 1990s, there has been a lot of work on algorithms to compute $\zeta_X(T)$ (and related objects) for various classes of algebraic varieties.

One original motivation came from cryptography, where it became necessary to compute orders of groups of points on elliptic curves over extremely large finite fields (e.g., $\mathbb{F}_{2^{256}}$). Subsequently, Jacobians of genus 2 curves were also needed.

However, there are plenty of mathematical reasons to be interested in such algorithms. One example from my work: investigating analogues of the Sato-Tate conjecture for genus 2 curves.

Nowadays, there is even some motivation from mathematical physics: arithmetic analogues of mirror symmetry.

Machine computation of zeta functions: motivation

Since the late 1990s, there has been a lot of work on algorithms to compute $\zeta_X(T)$ (and related objects) for various classes of algebraic varieties.

One original motivation came from cryptography, where it became necessary to compute orders of groups of points on elliptic curves over extremely large finite fields (e.g., $\mathbb{F}_{2^{256}}$). Subsequently, Jacobians of genus 2 curves were also needed.

However, there are plenty of mathematical reasons to be interested in such algorithms. One example from my work: investigating analogues of the Sato-Tate conjecture for genus 2 curves.

Nowadays, there is even some motivation from mathematical physics: arithmetic analogues of mirror symmetry.

Machine computation of zeta functions: motivation

Since the late 1990s, there has been a lot of work on algorithms to compute $\zeta_X(T)$ (and related objects) for various classes of algebraic varieties.

One original motivation came from cryptography, where it became necessary to compute orders of groups of points on elliptic curves over extremely large finite fields (e.g., $\mathbb{F}_{2^{256}}$). Subsequently, Jacobians of genus 2 curves were also needed.

However, there are plenty of mathematical reasons to be interested in such algorithms. One example from my work: investigating analogues of the Sato-Tate conjecture for genus 2 curves.

Nowadays, there is even some motivation from mathematical physics: arithmetic analogues of mirror symmetry.

Computation of zeta functions via étale cohomology

It is natural to try to use étale cohomology as the basis of algorithms for computing zeta functions.

One example is Schoof's algorithm for elliptic curves (circa 1985): compute the trace of Frobenius on ℓ-torsion for various small primes ℓ. With tweaks by Elkies and Atkin (early 1990s), this is quite practical.

Pila generalized Schoof's algorithm to abelian varieties. This is barely practical for genus 2 curves (Gaudry-Schost, 2010) and much more useful for genus 2 curves with real multiplication (Gaudry-Kohel-Smith, 2011).

Edixhoven's work on computing coefficients of modular forms (ongoing) is in a similar spirit.

It is unclear how to do anything more general. The essential difficulty seems to be handling étale cohomology in degree greater than 1.

Computation of zeta functions via étale cohomology

It is natural to try to use étale cohomology as the basis of algorithms for computing zeta functions.

One example is Schoof's algorithm for elliptic curves (circa 1985): compute the trace of Frobenius on ℓ-torsion for various small primes ℓ. With tweaks by Elkies and Atkin (early 1990s), this is quite practical.

Pila generalized Schoof's algorithm to abelian varieties. This is barely practical for genus 2 curves (Gaudry-Schost, 2010) and much more useful for genus 2 curves with real multiplication (Gaudry-Kohel-Smith, 2011).

Edixhoven's work on computing coefficients of modular forms (ongoing) is in a similar spirit.

It is unclear how to do anything more general. The essential difficulty seems to be handling étale cohomology in degree greater than 1.

Computation of zeta functions via étale cohomology

It is natural to try to use étale cohomology as the basis of algorithms for computing zeta functions.

One example is Schoof's algorithm for elliptic curves (circa 1985): compute the trace of Frobenius on ℓ-torsion for various small primes ℓ. With tweaks by Elkies and Atkin (early 1990s), this is quite practical.

Pila generalized Schoof's algorithm to abelian varieties. This is barely practical for genus 2 curves (Gaudry-Schost, 2010) and much more useful for genus 2 curves with real multiplication (Gaudry-Kohel-Smith, 2011).

Edixhoven's work on computing coefficients of modular forms (ongoing) is in a similar spirit.

It is unclear how to do anything more general. The essential difficulty seems to be handling étale cohomology in degree greater than 1.

Computation of zeta functions via étale cohomology

It is natural to try to use étale cohomology as the basis of algorithms for computing zeta functions.

One example is Schoof's algorithm for elliptic curves (circa 1985): compute the trace of Frobenius on ℓ-torsion for various small primes ℓ. With tweaks by Elkies and Atkin (early 1990s), this is quite practical.

Pila generalized Schoof's algorithm to abelian varieties. This is barely practical for genus 2 curves (Gaudry-Schost, 2010) and much more useful for genus 2 curves with real multiplication (Gaudry-Kohel-Smith, 2011).

Edixhoven's work on computing coefficients of modular forms (ongoing) is in a similar spirit.

It is unclear how to do anything more general. The essential difficulty seems to be handling étale cohomology in degree greater than 1.

Computation of zeta functions via étale cohomology

It is natural to try to use étale cohomology as the basis of algorithms for computing zeta functions.

One example is Schoof's algorithm for elliptic curves (circa 1985): compute the trace of Frobenius on ℓ-torsion for various small primes ℓ. With tweaks by Elkies and Atkin (early 1990s), this is quite practical.

Pila generalized Schoof's algorithm to abelian varieties. This is barely practical for genus 2 curves (Gaudry-Schost, 2010) and much more useful for genus 2 curves with real multiplication (Gaudry-Kohel-Smith, 2011).

Edixhoven's work on computing coefficients of modular forms (ongoing) is in a similar spirit.

It is unclear how to do anything more general. The essential difficulty seems to be handling étale cohomology in degree greater than 1.

Zeta functions via p-adic analysis

For elliptic curves over finite fields of *small characteristic*, several practical p-analytic methods were discovered for computing zeta functions, including Satoh's canonical lift method and Mestre's arithmetic-geometric mean iteration (both circa 1998). These do not generalize very far beyond elliptic curves, though.

Lauder and Wan (2000) described an algorithm based on Dwork's proof of rationality, applicable to *any* algebraic variety whatsoever! However, this is currently believed to be impractical.

Most practical algorithms for zeta functions via p-adic analysis go through the relationship between p-adic Weil cohomologies (crystalline, rigid) with algebraic de Rham cohomology. After my work on hyperelliptic curves with p odd (2001), much progress has been made by Denef and his *Belgian school* (Vercauteren, Castryck, Hubrechts, Tuitman).

Zeta functions via p-adic analysis

For elliptic curves over finite fields of *small characteristic*, several practical p-analytic methods were discovered for computing zeta functions, including Satoh's canonical lift method and Mestre's arithmetic-geometric mean iteration (both circa 1998). These do not generalize very far beyond elliptic curves, though.

Lauder and Wan (2000) described an algorithm based on Dwork's proof of rationality, applicable to *any* algebraic variety whatsoever! However, this is currently believed to be impractical.

Most practical algorithms for zeta functions via p-adic analysis go through the relationship between p-adic Weil cohomologies (crystalline, rigid) with algebraic de Rham cohomology. After my work on hyperelliptic curves with p odd (2001), much progress has been made by Denef and his *Belgian school* (Vercauteren, Castryck, Hubrechts, Tuitman).

Zeta functions via p-adic analysis

For elliptic curves over finite fields of *small characteristic*, several practical p-analytic methods were discovered for computing zeta functions, including Satoh's canonical lift method and Mestre's arithmetic-geometric mean iteration (both circa 1998). These do not generalize very far beyond elliptic curves, though.

Lauder and Wan (2000) described an algorithm based on Dwork's proof of rationality, applicable to *any* algebraic variety whatsoever! However, this is currently believed to be impractical.

Most practical algorithms for zeta functions via p-adic analysis go through the relationship between p-adic Weil cohomologies (crystalline, rigid) with algebraic de Rham cohomology. After my work on hyperelliptic curves with p odd (2001), much progress has been made by Denef and his *Belgian school* (Vercauteren, Castryck, Hubrechts, Tuitman).

Going beyond curves

While algorithms for p-adic cohomology are not intrinsically limited to curves, it seems difficult to get *practical* algorithms in higher dimension.

One approach is Lauder's *deformation method*, using Picard-Fuchs equations (i.e., Gauss-Manin connections), but little progress has been made in making this practical except for curves (Hubrechts).

Abbott, K, Roe (2007) considered the example of smooth projective hypersurfaces, based on Griffiths's description of the algebraic de Rham cohomology of same, but this was not very practical either.

What we describe today is a variant of AKR based on the principle of *controlled reduction* in algebraic de Rham cohomology. This turns out to be much more practical. In the process, we generalize to hypersurfaces in projective toric varieties.

Going beyond curves

While algorithms for p-adic cohomology are not intrinsically limited to curves, it seems difficult to get *practical* algorithms in higher dimension.

One approach is Lauder's *deformation method*, using Picard-Fuchs equations (i.e., Gauss-Manin connections), but little progress has been made in making this practical except for curves (Hubrechts).

Abbott, K, Roe (2007) considered the example of smooth projective hypersurfaces, based on Griffiths's description of the algebraic de Rham cohomology of same, but this was not very practical either.

What we describe today is a variant of AKR based on the principle of *controlled reduction* in algebraic de Rham cohomology. This turns out to be much more practical. In the process, we generalize to hypersurfaces in projective toric varieties.

Going beyond curves

While algorithms for p-adic cohomology are not intrinsically limited to curves, it seems difficult to get *practical* algorithms in higher dimension.

One approach is Lauder's *deformation method*, using Picard-Fuchs equations (i.e., Gauss-Manin connections), but little progress has been made in making this practical except for curves (Hubrechts).

Abbott, K, Roe (2007) considered the example of smooth projective hypersurfaces, based on Griffiths's description of the algebraic de Rham cohomology of same, but this was not very practical either.

What we describe today is a variant of AKR based on the principle of *controlled reduction* in algebraic de Rham cohomology. This turns out to be much more practical. In the process, we generalize to hypersurfaces in projective toric varieties.

Going beyond curves

While algorithms for p-adic cohomology are not intrinsically limited to curves, it seems difficult to get *practical* algorithms in higher dimension.

One approach is Lauder's *deformation method*, using Picard-Fuchs equations (i.e., Gauss-Manin connections), but little progress has been made in making this practical except for curves (Hubrechts).

Abbott, K, Roe (2007) considered the example of smooth projective hypersurfaces, based on Griffiths's description of the algebraic de Rham cohomology of same, but this was not very practical either.

What we describe today is a variant of AKR based on the principle of *controlled reduction* in algebraic de Rham cohomology. This turns out to be much more practical. In the process, we generalize to hypersurfaces in projective toric varieties.

Contents

Polarized toric varieties

Let Δ be a convex lattice polytope in $\mathbb{Z}^n$ not contained in any hyperplane.

Let P_d be the free R-module on $d\Delta \cap \mathbb{Z}^n$, and put $P = \oplus_{d=0}^{\infty} P_d$. Then $\mathrm{Proj}(P)$ is a projective normal toric variety over R carrying an ample torus-equivariant line bundle (and all such data arise this way).

Running example: for Δ equal to the simplex with vertices $0, \mathbf{e}_1, \dots, \mathbf{e}_n$, we get projective space. For $\mathbf{v} = c_1\mathbf{e}_1 + \cdots + c_n\mathbf{e}_n \in d\Delta \cap \mathbb{Z}^n$, identify the class $[\mathbf{v}] \in P_d$ with the homogeneous polynomial $x_0^{d-c_1-\cdots-c_n} x_1^{c_1} \cdots x_n^{c_n}$.

One may need to consider other examples (e.g., weighted projective spaces and products thereof) to pick up cases of interest (e.g., K3 surfaces with given Picard number, certain families of Calabi-Yau threefolds).

Polarized toric varieties

Let Δ be a convex lattice polytope in $\mathbb{Z}^n$ not contained in any hyperplane.

Let P_d be the free R-module on $d\Delta \cap \mathbb{Z}^n$, and put $P = \oplus_{d=0}^{\infty} P_d$. Then $\mathrm{Proj}(P)$ is a projective normal toric variety over R carrying an ample torus-equivariant line bundle (and all such data arise this way).

Running example: for Δ equal to the simplex with vertices $0, \mathbf{e}_1, \dots, \mathbf{e}_n$, we get projective space. For $\mathbf{v} = c_1\mathbf{e}_1 + \cdots + c_n\mathbf{e}_n \in d\Delta \cap \mathbb{Z}^n$, identify the class $[\mathbf{v}] \in P_d$ with the homogeneous polynomial $x_0^{d-c_1-\cdots-c_n} x_1^{c_1} \cdots x_n^{c_n}$.

One may need to consider other examples (e.g., weighted projective spaces and products thereof) to pick up cases of interest (e.g., K3 surfaces with given Picard number, certain families of Calabi-Yau threefolds).

Polarized toric varieties

Let Δ be a convex lattice polytope in $\mathbb{Z}^n$ not contained in any hyperplane.

Let P_d be the free R-module on $d\Delta \cap \mathbb{Z}^n$, and put $P = \oplus_{d=0}^{\infty} P_d$. Then $\mathrm{Proj}(P)$ is a projective normal toric variety over R carrying an ample torus-equivariant line bundle (and all such data arise this way).

Running example: for Δ equal to the simplex with vertices $0, \mathbf{e}_1, \dots, \mathbf{e}_n$, we get projective space. For $\mathbf{v} = c_1\mathbf{e}_1 + \cdots + c_n\mathbf{e}_n \in d\Delta \cap \mathbb{Z}^n$, identify the class $[\mathbf{v}] \in P_d$ with the homogeneous polynomial $x_0^{d-c_1-\cdots-c_n} x_1^{c_1} \cdots x_n^{c_n}$.

One may need to consider other examples (e.g., weighted projective spaces and products thereof) to pick up cases of interest (e.g., K3 surfaces with given Picard number, certain families of Calabi-Yau threefolds).

Polarized toric varieties

Let Δ be a convex lattice polytope in $\mathbb{Z}^n$ not contained in any hyperplane.

Let P_d be the free R-module on $d\Delta \cap \mathbb{Z}^n$, and put $P = \oplus_{d=0}^{\infty} P_d$. Then $\mathrm{Proj}(P)$ is a projective normal toric variety over R carrying an ample torus-equivariant line bundle (and all such data arise this way).

Running example: for Δ equal to the simplex with vertices $0, \mathbf{e}_1, \dots, \mathbf{e}_n$, we get projective space. For $\mathbf{v} = c_1\mathbf{e}_1 + \cdots + c_n\mathbf{e}_n \in d\Delta \cap \mathbb{Z}^n$, identify the class $[\mathbf{v}] \in P_d$ with the homogeneous polynomial $x_0^{d-c_1-\cdots-c_n} x_1^{c_1} \cdots x_n^{c_n}$.

One may need to consider other examples (e.g., weighted projective spaces and products thereof) to pick up cases of interest (e.g., K3 surfaces with given Picard number, certain families of Calabi-Yau threefolds).

Nondegeneracy of toric hypersurfaces

Choose $f \in P_d$ for some $d > 0$. We say f is *nondegenerate* if the hypersurface cut out by f has transversal intersection with each torus in the natural stratification of $\mathrm{Proj}(P)$.

For each $\lambda \in (\mathbb{Z}^n)^\vee$, define the derivation ∂_λ on P taking $[\mathbf{v}]$ to $\lambda(\mathbf{v})[\mathbf{v}]$ for $\mathbf{v} \in d\Delta \cap \mathbb{Z}^n$. For projective space, the standard basis of $(\mathbb{Z}^n)^\vee$ gives rise to the derivations $x_1 \frac{\partial}{\partial x_1}, \dots, x_n \frac{\partial}{\partial x_n}$.

The *toric Jacobian ideal* I_f in P is generated by f and all of the $\partial_\lambda(f)$. Then f is nondegenerate if and only if I_f is *irrelevant*, i.e., if there exists α such that $P_\beta \subseteq I_f$ for all $\beta \geq \alpha$. Using that $\mathrm{Spec}(P) \to \mathrm{Spec}(R)$ is Cohen-Macaulay, one can determine α in terms of Δ, d; for example, for projective space, we may take $\alpha = (n+1)(d-1)+1$.

Nondegeneracy of toric hypersurfaces

Choose $f \in P_d$ for some $d > 0$. We say f is *nondegenerate* if the hypersurface cut out by f has transversal intersection with each torus in the natural stratification of $\mathrm{Proj}(P)$.

For each $\lambda \in (\mathbb{Z}^n)^\vee$, define the derivation ∂_λ on P taking $[\mathbf{v}]$ to $\lambda(\mathbf{v})[\mathbf{v}]$ for $\mathbf{v} \in d\Delta \cap \mathbb{Z}^n$. For projective space, the standard basis of $(\mathbb{Z}^n)^\vee$ gives rise to the derivations $x_1 \frac{\partial}{\partial x_1}, \dots, x_n \frac{\partial}{\partial x_n}$.

The *toric Jacobian ideal* I_f in P is generated by f and all of the $\partial_\lambda(f)$. Then f is nondegenerate if and only if I_f is *irrelevant*, i.e., if there exists α such that $P_\beta \subseteq I_f$ for all $\beta \geq \alpha$. Using that $\mathrm{Spec}(P) \to \mathrm{Spec}(R)$ is Cohen-Macaulay, one can determine α in terms of Δ, d; for example, for projective space, we may take $\alpha = (n+1)(d-1)+1$.

Nondegeneracy of toric hypersurfaces

Choose $f \in P_d$ for some $d > 0$. We say f is *nondegenerate* if the hypersurface cut out by f has transversal intersection with each torus in the natural stratification of $\mathrm{Proj}(P)$.

For each $\lambda \in (\mathbb{Z}^n)^\vee$, define the derivation ∂_λ on P taking $[\mathbf{v}]$ to $\lambda(\mathbf{v})[\mathbf{v}]$ for $\mathbf{v} \in d\Delta \cap \mathbb{Z}^n$. For projective space, the standard basis of $(\mathbb{Z}^n)^\vee$ gives rise to the derivations $x_1 \frac{\partial}{\partial x_1}, \dots, x_n \frac{\partial}{\partial x_n}$.

The *toric Jacobian ideal* I_f in P is generated by f and all of the $\partial_\lambda(f)$. Then f is nondegenerate if and only if I_f is *irrelevant*, i.e., if there exists α such that $P_\beta \subseteq I_f$ for all $\beta \geq \alpha$. Using that $\mathrm{Spec}(P) \to \mathrm{Spec}(R)$ is Cohen-Macaulay, one can determine α in terms of Δ, d; for example, for projective space, we may take $\alpha = (n+1)(d-1)+1$.

de Rham cohomology of nondegenerate hypersurfaces

Suppose the base ring R is a field of characteristic 0 and that $f \in P_d$ is nondegenerate. Put $S = \cup_{i=0}^{\infty} f^{-i} P_{id}$; this is the coordinate ring of the nonzero locus U_f of f in $\mathrm{Proj}(P)$.

Let Z be the toric boundary of $\mathrm{Proj}(P)$ (i.e., the complement of the embedded torus $\mathrm{Spec}\, R[\mathbb{Z}^n]$). By Deligne, the algebraic de Rham cohomology of $U_f - Z$ is equal to the logarithmic de Rham cohomology of U_f for the log-structure defined by Z, i.e., the cohomology of the complex $\Omega^{\cdot}$ in which Ω^i is the free S-module on the generators

$$\mathrm{dlog}[\mathbf{e}_{j_1}] \wedge \cdots \wedge \mathrm{dlog}[\mathbf{e}_{j_i}] \qquad (1 \leq j_1 < \cdots < j_i \leq n)$$

with the usual exterior derivative.

de Rham cohomology of nondegenerate hypersurfaces

Suppose the base ring R is a field of characteristic 0 and that $f \in P_d$ is nondegenerate. Put $S = \cup_{i=0}^{\infty} f^{-i} P_{id}$; this is the coordinate ring of the nonzero locus U_f of f in $\mathrm{Proj}(P)$.

Let Z be the toric boundary of $\mathrm{Proj}(P)$ (i.e., the complement of the embedded torus $\mathrm{Spec}\, R[\mathbb{Z}^n]$). By Deligne, the algebraic de Rham cohomology of $U_f - Z$ is equal to the logarithmic de Rham cohomology of U_f for the log-structure defined by Z, i.e., the cohomology of the complex $\Omega^{\cdot}$ in which Ω^i is the free S-module on the generators

$$\mathrm{dlog}[\mathbf{e}_{j_1}] \wedge \cdots \wedge \mathrm{dlog}[\mathbf{e}_{j_i}] \qquad (1 \leq j_1 < \cdots < j_i \leq n)$$

with the usual exterior derivative.

de Rham cohomology: explicit generators and relations

The only cohomology which is interesting (i.e., not explained by the cohomology of $\mathrm{Proj}(P) - Z$) is in degree n, i.e., the cokernel H^n of $d : \Omega^{n-1} \to \Omega^n$. Put $\omega = \mathrm{dlog}[\mathbf{e}_1] \wedge \cdots \wedge \mathrm{dlog}[\mathbf{e}_n]$; then Ω^n is free on the generator ω, and H^n is the quotient by the R-submodule generated by

$$\frac{\partial_\lambda(g)}{f^m}\omega - m\frac{g\partial_\lambda(f)}{f^{m+1}}\omega$$

for each $\lambda \in (\mathbb{Z}^n)^\vee$, each nonnegative integer m, and each $g \in P_{md}$.

The link to p-adic cohomology

Now take $R = W(\mathbb{F}_q)$ for $\mathbb{F}_q$ a finite field of characteristic p. (That is, R is the finite étale extension of $\mathbb{Z}_p$ with residue field $\mathbb{F}_q$.) If we compute H^n over $R[p^{-1}]$, the result "is" the Monsky-Washnitzer cohomology (p-adic rigid cohomology) of the affine scheme $U_f - Z$ defined over $R/(p)$.

What this means explicitly is that there is a particular linear transformation of H^n (Frobenius) whose characteristic polynomial determines (the interesting factor of) the zeta function of $U_f - Z$. This in turn determines the zeta function of the zero locus of f on the big torus $\mathrm{Proj}(P) - Z$; one can repeat the construction to get the zeta functions of the zero loci on the boundary tori.

The link to p-adic cohomology

Now take $R = W(\mathbb{F}_q)$ for $\mathbb{F}_q$ a finite field of characteristic p. (That is, R is the finite étale extension of $\mathbb{Z}_p$ with residue field $\mathbb{F}_q$.) If we compute H^n over $R[p^{-1}]$, the result "is" the Monsky-Washnitzer cohomology (p-adic rigid cohomology) of the affine scheme $U_f - Z$ defined over $R/(p)$.

What this means explicitly is that there is a particular linear transformation of H^n (Frobenius) whose characteristic polynomial determines (the interesting factor of) the zeta function of $U_f - Z$. This in turn determines the zeta function of the zero locus of f on the big torus $\mathrm{Proj}(P) - Z$; one can repeat the construction to get the zeta functions of the zero loci on the boundary tori.

Frobenius in explicit form

In fact, the Frobenius map on H^n is quite explicit!

Although the endomorphism $\Phi : P \to P$ taking $[\mathbf{v}]$ to $[q\mathbf{v}] = [\mathbf{v}]^q$ does not extend to S, it does extend to a certain p-adic completion of S. We may formally extend Φ to differentials; given an element of H^n represented by $g\omega/f^m$, its image under Φ is the infinite sum

$$\begin{aligned}\frac{q^n\Phi(g)\omega}{\Phi(f)^m} &= \frac{q^n\Phi(g)\omega}{f^{qm}}\left(\frac{\Phi(f)}{f^q}\right)^{-m} \\ &= \frac{q^n\Phi(g)\omega}{f^{qm}}\sum_{i=0}^{\infty}\binom{-m}{i}\left(\frac{\Phi(f)-f^q}{f^q}\right)^i.\end{aligned}$$

(Note that $\Phi(\omega) = q^n\omega$ and that $\Phi(f) - f^q$ is divisible by p.)

Frobenius in explicit form

In fact, the Frobenius map on H^n is quite explicit!

Although the endomorphism $\Phi : P \to P$ taking $[\mathbf{v}]$ to $[q\mathbf{v}] = [\mathbf{v}]^q$ does not extend to S, it does extend to a certain p-adic completion of S. We may formally extend Φ to differentials; given an element of H^n represented by $g\omega/f^m$, its image under Φ is the infinite sum

$$\begin{aligned}\frac{q^n\Phi(g)\omega}{\Phi(f)^m} &= \frac{q^n\Phi(g)\omega}{f^{qm}}\left(\frac{\Phi(f)}{f^q}\right)^{-m} \\ &= \frac{q^n\Phi(g)\omega}{f^{qm}}\sum_{i=0}^{\infty}\binom{-m}{i}\left(\frac{\Phi(f)-f^q}{f^q}\right)^i.\end{aligned}$$

(Note that $\Phi(\omega) = q^n\omega$ and that $\Phi(f) - f^q$ is divisible by p.)

Contents

Computing in de Rham cohomology: the plan

Let's suppose again that R is a field and that $f \in P_d$ is nondegenerate. Using the relations defining H^n, it is not difficult to write down elements of Ω^n which project to a basis of H^n. What we now need is a way to express an arbitrary element of Ω^n as a linear combination of basis vectors plus a relation. We will typically start with a form looking like $g\omega/f^m$ with m large, so we think of this last step as *reduction of the pole order* along f.

If we can do that, then we get an algorithm for computing zeta functions of nondegenerate toric hypersurfaces using p-adic cohomology: starting with our complex over $W(\mathbb{F}_q)$, write down the action of Frobenius on basis representatives, then reduce each term in the resulting infinite sums (after inverting p).

For prescribed p-adic accuracy, we need only finitely many terms. How many? That's a delicate question which I neglect here.

Computing in de Rham cohomology: the plan

Let's suppose again that R is a field and that $f \in P_d$ is nondegenerate. Using the relations defining H^n, it is not difficult to write down elements of Ω^n which project to a basis of H^n. What we now need is a way to express an arbitrary element of Ω^n as a linear combination of basis vectors plus a relation. We will typically start with a form looking like $g\omega/f^m$ with m large, so we think of this last step as *reduction of the pole order* along f.

If we can do that, then we get an algorithm for computing zeta functions of nondegenerate toric hypersurfaces using p-adic cohomology: starting with our complex over $W(\mathbb{F}_q)$, write down the action of Frobenius on basis representatives, then reduce each term in the resulting infinite sums (after inverting p).

For prescribed p-adic accuracy, we need only finitely many terms. How many? That's a delicate question which I neglect here.

Computing in de Rham cohomology: the plan

Let's suppose again that R is a field and that $f \in P_d$ is nondegenerate. Using the relations defining H^n, it is not difficult to write down elements of Ω^n which project to a basis of H^n. What we now need is a way to express an arbitrary element of Ω^n as a linear combination of basis vectors plus a relation. We will typically start with a form looking like $g\omega/f^m$ with m large, so we think of this last step as *reduction of the pole order* along f.

If we can do that, then we get an algorithm for computing zeta functions of nondegenerate toric hypersurfaces using p-adic cohomology: starting with our complex over $W(\mathbb{F}_q)$, write down the action of Frobenius on basis representatives, then reduce each term in the resulting infinite sums (after inverting p).

For prescribed p-adic accuracy, we need only finitely many terms. How many? That's a delicate question which I neglect here.

The difficulty: too many terms

To reduce the pole order of $g\omega/f^m$ from m to $m-1$, we must write g as a P-linear combination of f and its partial derivatives. One might use Gröbner basis methods as implemented in a standard computer algebra package (e.g., Singular or Magma). This gives uncontrollable asymptotics, so it is better to find these representations using direct linear algebra.

There remains a serious problem: we typically start with g being rather sparse, but an ill-conceived reduction algorithm will produce *dense* polynomials. This typically leads to a factor of p^d in time and space complexity of the resulting algorithms, which limits practicality. One must really limit this to p^1 instead!

The difficulty: too many terms

To reduce the pole order of $g\omega/f^m$ from m to $m-1$, we must write g as a P-linear combination of f and its partial derivatives. One might use Gröbner basis methods as implemented in a standard computer algebra package (e.g., Singular or Magma). This gives uncontrollable asymptotics, so it is better to find these representations using direct linear algebra.

There remains a serious problem: we typically start with g being rather sparse, but an ill-conceived reduction algorithm will produce *dense* polynomials. This typically leads to a factor of p^d in time and space complexity of the resulting algorithms, which limits practicality. One must really limit this to p^1 instead!

The fix: controlled reduction

The solution is to exhibit a reduction procedure that preserves sparsity, in terms of the integer α we chose for which $P_\beta \subseteq I_f$ for all $\beta \geq \alpha$.

Theorem (Controlled reduction)

Suppose $\mathbb{Q} \subseteq R$. Choose an integer β with $\beta + d \geq \alpha$, an integer m with $md \geq \beta$, and monomials $\mu \in P_d, \nu \in P_{md-\beta}$. We can then find R-linear maps $R_0, R_1 : P_\beta \to P_\beta$ such that for any $x \in P_\beta, j \geq 0$,

$$\frac{x\mu^{j+1}\nu}{f^{m+j+1}}\omega \equiv (m+j)^{-1}(R_0(x) + jR_1(x))\frac{\mu^j\nu}{f^{m+j}}\omega \quad \text{in } H^n.$$

The point is that $(m+j)^{-1}(R_0(x) + jR_1(x))$ is again in P_β. Hence starting with $g\omega/f^m$ with m large and g sparse, we can write g as a linear combination of a few terms, each equal to a high power of some monomial μ times a small cofactor. We then do controlled reduction to get some small terms, which we resolve by direct linear algebra.

The fix: controlled reduction

The solution is to exhibit a reduction procedure that preserves sparsity, in terms of the integer α we chose for which $P_\beta \subseteq I_f$ for all $\beta \geq \alpha$.

Theorem (Controlled reduction)

Suppose $\mathbb{Q} \subseteq R$. Choose an integer β with $\beta + d \geq \alpha$, an integer m with $md \geq \beta$, and monomials $\mu \in P_d, \nu \in P_{md-\beta}$. We can then find R-linear maps $R_0, R_1 : P_\beta \to P_\beta$ such that for any $x \in P_\beta, j \geq 0$,

$$\frac{x\mu^{j+1}\nu}{f^{m+j+1}}\omega \equiv (m+j)^{-1}(R_0(x) + jR_1(x))\frac{\mu^j\nu}{f^{m+j}}\omega \quad \text{in } H^n.$$

The point is that $(m+j)^{-1}(R_0(x) + jR_1(x))$ is again in P_β. Hence starting with $g\omega/f^m$ with m large and g sparse, we can write g as a linear combination of a few terms, each equal to a high power of some monomial μ times a small cofactor. We then do controlled reduction to get some small terms, which we resolve by direct linear algebra.

The fix: controlled reduction

The solution is to exhibit a reduction procedure that preserves sparsity, in terms of the integer α we chose for which $P_\beta \subseteq I_f$ for all $\beta \geq \alpha$.

Theorem (Controlled reduction)

Suppose $\mathbb{Q} \subseteq R$. Choose an integer β with $\beta + d \geq \alpha$, an integer m with $md \geq \beta$, and monomials $\mu \in P_d, \nu \in P_{md-\beta}$. We can then find R-linear maps $R_0, R_1 : P_\beta \to P_\beta$ such that for any $x \in P_\beta, j \geq 0$,

$$\frac{x\mu^{j+1}\nu}{f^{m+j+1}}\omega \equiv (m+j)^{-1}(R_0(x) + jR_1(x))\frac{\mu^j\nu}{f^{m+j}}\omega \quad \text{in } H^n.$$

The point is that $(m+j)^{-1}(R_0(x) + jR_1(x))$ is again in P_β. Hence starting with $g\omega/f^m$ with m large and g sparse, we can write g as a linear combination of a few terms, each equal to a high power of some monomial μ times a small cofactor. We then do controlled reduction to get some small terms, which we resolve by direct linear algebra.

Proof of controlled reduction

By the choice of β, there exist R-linear maps $\pi_0, \dots, \pi_n : P_\beta \to P_\beta$ with $\mu x = \pi_0(x) f + \sum_{i=1}^n \pi_i \partial_{\mathbf{e}_i^*}(f)$. Then take

$$R_0(x) = m\pi_0(x) + \sum_{i=1}^n (\partial_{\mathbf{e}_i^*} + \mathbf{e}_i^*(\nu))(\pi_i(x))$$

$$R_1(x) = \pi_0(x) + \sum_{i=1}^n \mathbf{e}_i^*(\mu)\pi_i(x).$$

We then have as desired:

$$\frac{x\mu^{j+1}\nu}{f^{m+j+1}}\omega \equiv (m+j)^{-1}(R_0(x) + jR_1(x))\frac{\mu^j\nu}{f^{m+j}}\omega.$$

Contents

Experimental results

So far, we have only implemented this for projective space, and only in Sage (i.e., not in any optimized fashion).

Nonetheless, we computed the zeta function of a random quartic surface in $\mathbb{P}^3$ over $\mathbb{F}_{10^3+9}$ in two CPU-days, and over $\mathbb{F}_{10^6+3}$ in about 20 CPU-days. It is easy to parallelize, and anyway an optimized version should be many times faster!

By contrast, the original AKR algorithm, implemented in Magma, was unable to handle quartic surfaces over $\mathbb{F}_p$ except for $p \leq 19$.

Experimental results

So far, we have only implemented this for projective space, and only in Sage (i.e., not in any optimized fashion).

Nonetheless, we computed the zeta function of a random quartic surface in $\mathbb{P}^3$ over $\mathbb{F}_{10^3+9}$ in two CPU-days, and over $\mathbb{F}_{10^6+3}$ in about 20 CPU-days. It is easy to parallelize, and anyway an optimized version should be many times faster!

By contrast, the original AKR algorithm, implemented in Magma, was unable to handle quartic surfaces over $\mathbb{F}_p$ except for $p \leq 19$.

Experimental results

So far, we have only implemented this for projective space, and only in Sage (i.e., not in any optimized fashion).

Nonetheless, we computed the zeta function of a random quartic surface in $\mathbb{P}^3$ over $\mathbb{F}_{10^3+9}$ in two CPU-days, and over $\mathbb{F}_{10^6+3}$ in about 20 CPU-days. It is easy to parallelize, and anyway an optimized version should be many times faster!

By contrast, the original AKR algorithm, implemented in Magma, was unable to handle quartic surfaces over $\mathbb{F}_p$ except for $p \leq 19$.

Room for improvement: p^1 to $p^{1/2}$

Previously, Harvey improved the dependence on p in my original algorithm for hyperelliptic curves from p^1 to $p^{1/2}$. This uses a technique of the Chudnovskys to accelerate the computation of a linear recurrence with polynomial coefficients by "giant-stepping": instead of taking p individual recursion steps, one takes $\sqrt{p}$ batches of steps of length $\sqrt{p}$.

Controlled reduction makes it possible to do this for toric hypersurfaces too, but we haven't tried yet, so it is unclear how much this will help. For very small p, it might make things worse.

Room for improvement: p^1 to $p^{1/2}$

Previously, Harvey improved the dependence on p in my original algorithm for hyperelliptic curves from p^1 to $p^{1/2}$. This uses a technique of the Chudnovskys to accelerate the computation of a linear recurrence with polynomial coefficients by "giant-stepping": instead of taking p individual recursion steps, one takes $\sqrt{p}$ batches of steps of length $\sqrt{p}$.

Controlled reduction makes it possible to do this for toric hypersurfaces too, but we haven't tried yet, so it is unclear how much this will help. For very small p, it might make things worse.

Partially nondegenerate hypersurfaces

One can also weaken the nondegenerate condition somewhat, by forcing controlled reduction in particular directions. For instance, in projective space, one can handle arbitrary smooth hypersurfaces (having arbitrarily bad intersections with the toric boundary) as soon as $d \geq n + 1$.

We wrote down a generalization to toric varieties can be written down, but it is somewhat complicated to use. For instance, it is unclear how to find the analogue of α, particularly because this may depend on p. For instance, in the case of projective space, there is trouble when $p|d$ because the Euler relation degenerates (creating a syzygy among the partial derivatives, as observed first by Beauville).

Partially nondegenerate hypersurfaces

One can also weaken the nondegenerate condition somewhat, by forcing controlled reduction in particular directions. For instance, in projective space, one can handle arbitrary smooth hypersurfaces (having arbitrarily bad intersections with the toric boundary) as soon as $d \geq n+1$.

We wrote down a generalization to toric varieties can be written down, but it is somewhat complicated to use. For instance, it is unclear how to find the analogue of α, particularly because this may depend on p. For instance, in the case of projective space, there is trouble when $p|d$ because the Euler relation degenerates (creating a syzygy among the partial derivatives, as observed first by Beauville).