Zeta functions of algebraic varieties: theory and algorithms

Kiran S. Kedlaya

Department of Mathematics, University of California San Diego
kedlaya@ucsd.edu

These slides can be downloaded from https://kskedlaya.org/slides/.

Number theory meeting on Ramanujan's 135th birth year
Indian Institute of Technology, Kanpur
February 28, 2023

Supported by NSF (grant DMS-2053473), UC San Diego (Warschawski Professorship), and the Society of p-adicts.

The UC San Diego campus sits on unceded ancestral land of the Kumeyaay Nation.

Contents

Algebraic varieties over finite fields

Let $\mathbb{F}_q$ denote a finite field. We will consider **algebraic varieties** over $\mathbb{F}_q$, such as the following.

- An **affine algebraic variety** "is" the set of solutions of some system of polynomial equations

$$f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0$$

  in the n-dimensional affine space $\mathbb{A}^n_{\mathbb{F}_q}$.
- A **projective algebraic variety** "is" the set of solutions of some system of **homogeneous** polynomial equations

$$f_1(x_0, \ldots, x_n) = \cdots = f_m(x_0, \ldots, x_n) = 0$$

  in the n-dimensional projective space $\mathbb{P}^n_{\mathbb{F}_q}$ (= the set of one-dimensional subspaces of $\mathbb{A}^{n+1}_{\mathbb{F}_q}$).

In both cases, we remember the (ideal generated by) the defining equations, so that we can also compute solutions over extension fields $\mathbb{F}_{q^d}$.

Algebraic varieties over finite fields

Let $\mathbb{F}_q$ denote a finite field. We will consider **algebraic varieties** over $\mathbb{F}_q$, such as the following.

- An **affine algebraic variety** "is" the set of solutions of some system of polynomial equations

$$f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0$$

  in the n-dimensional affine space $\mathbb{A}^n_{\mathbb{F}_q}$.
- A **projective algebraic variety** "is" the set of solutions of some system of **homogeneous** polynomial equations

$$f_1(x_0, \ldots, x_n) = \cdots = f_m(x_0, \ldots, x_n) = 0$$

  in the n-dimensional projective space $\mathbb{P}^n_{\mathbb{F}_q}$ (= the set of one-dimensional subspaces of $\mathbb{A}^{n+1}_{\mathbb{F}_q}$).

In both cases, we remember the (ideal generated by) the defining equations, so that we can also compute solutions over extension fields $\mathbb{F}_{q^d}$.

Algebraic varieties over finite fields

Let $\mathbb{F}_q$ denote a finite field. We will consider **algebraic varieties** over $\mathbb{F}_q$, such as the following.

- An **affine algebraic variety** "is" the set of solutions of some system of polynomial equations

$$f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0$$

  in the n-dimensional affine space $\mathbb{A}^n_{\mathbb{F}_q}$.
- A **projective algebraic variety** "is" the set of solutions of some system of **homogeneous** polynomial equations

$$f_1(x_0, \ldots, x_n) = \cdots = f_m(x_0, \ldots, x_n) = 0$$

  in the n-dimensional projective space $\mathbb{P}^n_{\mathbb{F}_q}$ (= the set of one-dimensional subspaces of $\mathbb{A}^{n+1}_{\mathbb{F}_q}$).

In both cases, we remember the (ideal generated by) the defining equations, so that we can also compute solutions over extension fields $\mathbb{F}_{q^d}$.

Algebraic varieties over finite fields

Let $\mathbb{F}_q$ denote a finite field. We will consider **algebraic varieties** over $\mathbb{F}_q$, such as the following.

- An **affine algebraic variety** "is" the set of solutions of some system of polynomial equations

$$f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0$$

  in the n-dimensional affine space $\mathbb{A}^n_{\mathbb{F}_q}$.
- A **projective algebraic variety** "is" the set of solutions of some system of **homogeneous** polynomial equations

$$f_1(x_0, \ldots, x_n) = \cdots = f_m(x_0, \ldots, x_n) = 0$$

  in the n-dimensional projective space $\mathbb{P}^n_{\mathbb{F}_q}$ (= the set of one-dimensional subspaces of $\mathbb{A}^{n+1}_{\mathbb{F}_q}$).

In both cases, we remember the (ideal generated by) the defining equations, so that we can also compute solutions over extension fields $\mathbb{F}_{q^d}$.

The Hasse-Weil zeta function of a variety

Let X be an algebraic variety over a finite field $\mathbb{F}_q$. The **Hasse-Weil zeta function** is the power series

$$Z(X,T) = \exp\left(\sum_{n=1}^{\infty} \frac{T^n}{n} \#X(\mathbb{F}_{q^n})\right).$$

To see where this definition came from, substitute $T = q^{-s}$; then

$$Z(X,q^{-s}) = \prod_x (1-(q^{\deg(x)})^{-s})^{-1}$$

where x runs over Galois orbits of points of X over $\overline{\mathbb{F}}_q$ (an algebraic closure) and $\deg(x)$ is the size of the orbit.

This **Euler product** is quite closely analogous to the definition of the Riemann zeta function, or the Dedekind zeta function over a number field.

The Hasse-Weil zeta function of a variety

Let X be an algebraic variety over a finite field $\mathbb{F}_q$. The **Hasse-Weil zeta function** is the power series

$$Z(X, T) = \exp\left(\sum_{n=1}^{\infty} \frac{T^n}{n} \#X(\mathbb{F}_{q^n})\right).$$

To see where this definition came from, substitute $T = q^{-s}$; then

$$Z(X, q^{-s}) = \prod_x (1 - (q^{\deg(x)})^{-s})^{-1}$$

where x runs over Galois orbits of points of X over $\overline{\mathbb{F}}_q$ (an algebraic closure) and $\deg(x)$ is the size of the orbit.

This **Euler product** is quite closely analogous to the definition of the Riemann zeta function, or the Dedekind zeta function over a number field.

The Hasse-Weil zeta function of a variety

Let X be an algebraic variety over a finite field $\mathbb{F}_q$. The **Hasse-Weil zeta function** is the power series

$$Z(X,T) = \exp\left(\sum_{n=1}^{\infty} \frac{T^n}{n} \#X(\mathbb{F}_{q^n})\right).$$

To see where this definition came from, substitute $T = q^{-s}$; then

$$Z(X,q^{-s}) = \prod_x (1-(q^{\deg(x)})^{-s})^{-1}$$

where x runs over Galois orbits of points of X over $\overline{\mathbb{F}}_q$ (an algebraic closure) and $\deg(x)$ is the size of the orbit.

This **Euler product** is quite closely analogous to the definition of the Riemann zeta function, or the Dedekind zeta function over a number field.

Examples

For $X = \mathbb{P}^n$,

$$Z(X,T) = \frac{1}{(1-T)(1-qT)\cdots(1-q^nT)}.$$

For X an elliptic curve,

$$Z(X,T) = \frac{1-aT+qT^2}{(1-T)(1-qT)}$$

where $|a| \leq 2q^{1/2}$. Note that $\#X(\mathbb{F}_q) = q+1-a$.

For X a curve of genus g,

$$Z(X,T) = \frac{1+a_1T+\cdots+a_gT^g+qa_{g-1}T^{g+1}+\cdots+q^gT^{2g}}{(1-T)(1-qT)}.$$

Examples

For $X = \mathbb{P}^n$,

$$Z(X,T) = \frac{1}{(1-T)(1-qT)\cdots(1-q^nT)}.$$

For X an elliptic curve,

$$Z(X,T) = \frac{1-aT+qT^2}{(1-T)(1-qT)}$$

where $|a| \leq 2q^{1/2}$. Note that $\#X(\mathbb{F}_q) = q+1-a$.

For X a curve of genus g,

$$Z(X,T) = \frac{1+a_1T+\cdots+a_gT^g+qa_{g-1}T^{g+1}+\cdots+q^gT^{2g}}{(1-T)(1-qT)}.$$

Examples

For $X = \mathbb{P}^n$,

$$Z(X,T) = \frac{1}{(1-T)(1-qT)\cdots(1-q^nT)}.$$

For X an elliptic curve,

$$Z(X,T) = \frac{1-aT+qT^2}{(1-T)(1-qT)}$$

where $|a| \leq 2q^{1/2}$. Note that $\#X(\mathbb{F}_q) = q+1-a$.

For X a curve of genus g,

$$Z(X,T) = \frac{1+a_1T+\cdots+a_gT^g+qa_{g-1}T^{g+1}+\cdots+q^gT^{2g}}{(1-T)(1-qT)}.$$

The Weil conjectures

Based on a mix of conceptual and numerical evidence, Weil conjectured that:

- $Z(X,T) \in 1 + T\mathbb{Z}[[T]]$;
- $Z(X,T)$ represents a rational function in T;
- every zero or pole z of $Z(X,T)$ in $\mathbb{C}$ satisfies $|z| = q^{-i/2}$ for some $i \in \{0,\ldots,2d\}$, where d is the dimension* of X;
- if X is smooth† projective‡, $Z(X, q^{-d}T^{-1}) = T^* q^* Z(X,T)$.

These are all now theorems through work of various authors (Dwork, Grothendieck, Deligne, etc.).

*E.g., if X is an affine variety cut out by m equations in n variables, then $d \leq n - m$.

†This is defined using partial derivatives (Jacobian criterion).

‡If you know what "proper" means, you can replace "projective" here.

The Weil conjectures

Based on a mix of conceptual and numerical evidence, Weil conjectured that:

- $Z(X,T) \in 1 + T\mathbb{Z}[[T]]$;
- $Z(X,T)$ represents a rational function in T;
- every zero or pole z of $Z(X,T)$ in $\mathbb{C}$ satisfies $|z| = q^{-i/2}$ for some $i \in \{0,\ldots,2d\}$, where d is the dimension* of X;
- if X is smooth† projective‡, $Z(X, q^{-d}T^{-1}) = T^* q^* Z(X,T)$.

These are all now theorems through work of various authors (Dwork, Grothendieck, Deligne, etc.).

*E.g., if X is an affine variety cut out by m equations in n variables, then $d \leq n - m$.

†This is defined using partial derivatives (Jacobian criterion).

‡If you know what "proper" means, you can replace "projective" here.

The Weil conjectures

Based on a mix of conceptual and numerical evidence, Weil conjectured that:

- $Z(X,T) \in 1 + T\mathbb{Z}[[T]]$;
- $Z(X,T)$ represents a rational function in T;
- every zero or pole z of $Z(X,T)$ in $\mathbb{C}$ satisfies $|z| = q^{-i/2}$ for some $i \in \{0,\ldots,2d\}$, where d is the dimension* of X;
- if X is smooth† projective‡, $Z(X, q^{-d}T^{-1}) = T^* q^* Z(X,T)$.

These are all now theorems through work of various authors (Dwork, Grothendieck, Deligne, etc.).

*E.g., if X is an affine variety cut out by m equations in n variables, then $d \leq n - m$.

†This is defined using partial derivatives (Jacobian criterion).

‡If you know what "proper" means, you can replace "projective" here.

The Weil conjectures

Based on a mix of conceptual and numerical evidence, Weil conjectured that:

- $Z(X,T) \in 1 + T\mathbb{Z}[[T]]$;
- $Z(X,T)$ represents a rational function in T;
- every zero or pole z of $Z(X,T)$ in $\mathbb{C}$ satisfies $|z| = q^{-i/2}$ for some $i \in \{0,\ldots,2d\}$, where d is the dimension* of X;
- if X is smooth† projective‡, $Z(X, q^{-d}T^{-1}) = T^* q^* Z(X,T)$.

These are all now theorems through work of various authors (Dwork, Grothendieck, Deligne, etc.).

*E.g., if X is an affine variety cut out by m equations in n variables, then $d \leq n - m$.

†This is defined using partial derivatives (Jacobian criterion).

‡If you know what "proper" means, you can replace "projective" here.

The Weil conjectures

Based on a mix of conceptual and numerical evidence, Weil conjectured that:

- $Z(X,T) \in 1 + T\mathbb{Z}[[T]]$;
- $Z(X,T)$ represents a rational function in T;
- every zero or pole z of $Z(X,T)$ in $\mathbb{C}$ satisfies $|z| = q^{-i/2}$ for some $i \in \{0,\ldots,2d\}$, where d is the dimension* of X;
- if X is smooth† projective‡, $Z(X, q^{-d}T^{-1}) = T^* q^* Z(X,T)$.

These are all now theorems through work of various authors (Dwork, Grothendieck, Deligne, etc.).

*E.g., if X is an affine variety cut out by m equations in n variables, then $d \leq n - m$.

†This is defined using partial derivatives (Jacobian criterion).

‡If you know what "proper" means, you can replace "projective" here.

The Weil conjectures

Based on a mix of conceptual and numerical evidence, Weil conjectured that:

- $Z(X,T) \in 1 + T\mathbb{Z}[[T]]$;
- $Z(X,T)$ represents a rational function in T;
- every zero or pole z of $Z(X,T)$ in $\mathbb{C}$ satisfies $|z| = q^{-i/2}$ for some $i \in \{0,\ldots,2d\}$, where d is the dimension* of X;
- if X is smooth† projective‡, $Z(X, q^{-d}T^{-1}) = T^* q^* Z(X,T)$.

These are all now theorems through work of various authors (Dwork, Grothendieck, Deligne, etc.).

*E.g., if X is an affine variety cut out by m equations in n variables, then $d \leq n - m$.

†This is defined using partial derivatives (Jacobian criterion).

‡If you know what "proper" means, you can replace "projective" here.

A consequence for modular forms

A **modular form of weight** k is a holomorphic function $f(z)$ for $\mathrm{Real}(z) > 0$ satisfying (a growth condition plus) the functional equation

$$f\left(\frac{az+b}{cz+d}\right) = (cz+d)^k f(z)$$

for matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ over $\mathbb{Z}$ congruent to 1 modulo N (the **level**).

Since f is invariant under $z \mapsto z+N$, it has a Fourier expansion

$$f(z) = \sum_{n=0}^{\infty} a_{n/N}(f) q^{n/N}, \qquad q = e^{2\pi i z}.$$

A famous example with $k = 12, N = 1$ is the **discriminant**:

$$\Delta(z) = q \prod_{i=1}^{\infty} (1-q^i)^{24} = \sum_{n=1}^{\infty} \tau(n) q^n$$

where $\tau(n) \in \mathbb{Z}$ means Ramanujan's tau function.

A consequence for modular forms

A **modular form of weight** k is a holomorphic function $f(z)$ for $\mathrm{Real}(z) > 0$ satisfying (a growth condition plus) the functional equation

$$f\left(\frac{az+b}{cz+d}\right) = (cz+d)^k f(z)$$

for matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ over $\mathbb{Z}$ congruent to 1 modulo N (the **level**).

Since f is invariant under $z \mapsto z+N$, it has a Fourier expansion

$$f(z) = \sum_{n=0}^{\infty} a_{n/N}(f) q^{n/N}, \qquad q = e^{2\pi i z}.$$

A famous example with $k = 12, N = 1$ is the **discriminant**:

$$\Delta(z) = q \prod_{i=1}^{\infty} (1-q^i)^{24} = \sum_{n=1}^{\infty} \tau(n) q^n$$

where $\tau(n) \in \mathbb{Z}$ means Ramanujan's tau function.

A consequence for modular forms

A **modular form of weight** k is a holomorphic function $f(z)$ for $\mathrm{Real}(z) > 0$ satisfying (a growth condition plus) the functional equation

$$f\left(\frac{az+b}{cz+d}\right) = (cz+d)^k f(z)$$

for matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ over $\mathbb{Z}$ congruent to 1 modulo N (the **level**).

Since f is invariant under $z \mapsto z+N$, it has a Fourier expansion

$$f(z) = \sum_{n=0}^{\infty} a_{n/N}(f) q^{n/N}, \qquad q = e^{2\pi i z}.$$

A famous example with $k = 12, N = 1$ is the **discriminant**:

$$\Delta(z) = q \prod_{i=1}^{\infty} (1-q^i)^{24} = \sum_{n=1}^{\infty} \tau(n) q^n$$

where $\tau(n) \in \mathbb{Z}$ means Ramanujan's tau function.

Ramunujan's conjecture

Ramanujan conjectured that for p prime,

$$|\tau(p)| \leq 2p^{11/2}.$$

Deligne proved this by finding a smooth projective scheme[§] X of dimension 11 over $\mathbb{Z}$ such that two of the zeroes α_p, β_p of $Z(X_{\mathbb{F}_p}, T)$ with $|\alpha_p| = |\beta_p| = p^{-11/2}$ satisfy

$$\alpha_p^{-1} + \beta_p^{-1} = \tau(p).$$

Using similar logic one obtains an analogous bound on the Fourier coefficients of other modular forms (specifically cuspidal eigenforms), resolving the **Ramanujan–Petersson conjecture**.

[§]Not quite: it's a Deligne–Mumford stack, but never mind about the difference.

Ramunujan's conjecture

Ramanujan conjectured that for p prime,

$$|\tau(p)| \leq 2p^{11/2}.$$

Deligne proved this by finding a smooth projective scheme[§] X of dimension 11 over $\mathbb{Z}$ such that two of the zeroes α_p, β_p of $Z(X_{\mathbb{F}_p}, T)$ with $|\alpha_p| = |\beta_p| = p^{-11/2}$ satisfy

$$\alpha_p^{-1} + \beta_p^{-1} = \tau(p).$$

Using similar logic one obtains an analogous bound on the Fourier coefficients of other modular forms (specifically cuspidal eigenforms), resolving the **Ramanujan–Petersson conjecture**.

[§]Not quite: it's a Deligne–Mumford stack, but never mind about the difference.

Ramunujan's conjecture

Ramanujan conjectured that for p prime,

$$|\tau(p)| \leq 2p^{11/2}.$$

Deligne proved this by finding a smooth projective scheme[§] X of dimension 11 over $\mathbb{Z}$ such that two of the zeroes α_p, β_p of $Z(X_{\mathbb{F}_p}, T)$ with $|\alpha_p| = |\beta_p| = p^{-11/2}$ satisfy

$$\alpha_p^{-1} + \beta_p^{-1} = \tau(p).$$

Using similar logic one obtains an analogous bound on the Fourier coefficients of other modular forms (specifically cuspidal eigenforms), resolving the **Ramanujan–Petersson conjecture**.

[§]Not quite: it's a Deligne–Mumford stack, but never mind about the difference.

Contents

A finiteness statement

Theorem

Let X be the affine variety defined by $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$. Then there exists a (randomized) algorithm that on input of m, n, and $f_1, \ldots, f_m$, outputs a representation of $Z(X, T)$ as a rational function over $\mathbb{Q}$. (And similarly for projective varieties.)

Sketch of proof.

First give an effective upper bound on the degree of the numerator and denominator of $Z(X, T)$ in terms of m, n, and the degrees of the f_i (eg., by induction on n). Then enumerate the $\mathbb{F}_{q^i}$-valued points of X for enough values of i to pin down the power series. □

This algorithm is used in practice in some small cases, but is not asymptotically best possible.

A finiteness statement

Theorem

Let X be the affine variety defined by $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$. Then there exists a (randomized) algorithm that on input of m, n, and $f_1, \ldots, f_m$, outputs a representation of $Z(X, T)$ as a rational function over $\mathbb{Q}$. (And similarly for projective varieties.)

Sketch of proof.

First give an effective upper bound on the degree of the numerator and denominator of $Z(X, T)$ in terms of m, n, and the degrees of the f_i (eg., by induction on n). Then enumerate the $\mathbb{F}_{q^i}$-valued points of X for enough values of i to pin down the power series. □

This algorithm is used in practice in some small cases, but is not asymptotically best possible.

A finiteness statement

Theorem

Let X be the affine variety defined by $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$. Then there exists a (randomized) algorithm that on input of m, n, and $f_1, \ldots, f_m$, outputs a representation of $Z(X, T)$ as a rational function over $\mathbb{Q}$. (And similarly for projective varieties.)

Sketch of proof.

First give an effective upper bound on the degree of the numerator and denominator of $Z(X, T)$ in terms of m, n, and the degrees of the f_i (eg., by induction on n). Then enumerate the $\mathbb{F}_{q^i}$-valued points of X for enough values of i to pin down the power series. □

This algorithm is used in practice in some small cases, but is not asymptotically best possible.

Statement of the problem

Problem

Find an **efficient** *algorithm to compute $Z(X, T)$ from an explicit representation of an affine or projective variety X.*

We are interested in this question both **in theory** and **in practice**. In theory, we want to handle arbitrary X in polynomial time in the input/output length. For X affine defined by $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$, this means $\mathrm{poly}(\log q, m, \max(\deg(f_i))^n)$; this is open even if we fix n.

In practice, we often restrict to special cases of interest where we have more efficient methods than in the general case. We also quantify in different ways; e.g., for X a smooth projective curve it is natural to quantify over its **genus**. In this setting we have many useful results and implementations, but there are still many open directions.

Statement of the problem

Problem

Find an **efficient** *algorithm to compute $Z(X, T)$ from an explicit representation of an affine or projective variety X.*

We are interested in this question both **in theory** and **in practice**. In theory, we want to handle arbitrary X in polynomial time in the input/output length. For X affine defined by $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$, this means $\mathrm{poly}(\log q, m, \max(\deg(f_i))^n)$; this is open even if we fix n.

In practice, we often restrict to special cases of interest where we have more efficient methods than in the general case. We also quantify in different ways; e.g., for X a smooth projective curve it is natural to quantify over its **genus**. In this setting we have many useful results and implementations, but there are still many open directions.

Statement of the problem

Problem

Find an **efficient** *algorithm to compute $Z(X, T)$ from an explicit representation of an affine or projective variety X.*

We are interested in this question both **in theory** and **in practice**. In theory, we want to handle arbitrary X in polynomial time in the input/output length. For X affine defined by $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$, this means $\mathrm{poly}(\log q, m, \max(\deg(f_i))^n)$; this is open even if we fix n.

In practice, we often restrict to special cases of interest where we have more efficient methods than in the general case. We also quantify in different ways; e.g., for X a smooth projective curve it is natural to quantify over its **genus**. In this setting we have many useful results and implementations, but there are still many open directions.

A word of warning

One might be tempted to use a **sparse** input representation of X. E.g., for $q = 2$, one can translate an instance of 3-SAT into a problem of computing $\#X(\mathbb{F}_q)$, which is the T^1 coefficient of $Z(X, T)$.

However, in this example the output length of $Z(X, T)$ is **exponential** in the input length. Thus the NP-hardness of 3-SAT does **not** imply any hardness result for our problem!

A word of warning

One might be tempted to use a **sparse** input representation of X. E.g., for $q = 2$, one can translate an instance of 3-SAT into a problem of computing $\#X(\mathbb{F}_q)$, which is the T^1 coefficient of $Z(X, T)$.

However, in this example the output length of $Z(X, T)$ is **exponential** in the input length. Thus the NP-hardness of 3-SAT does **not** imply any hardness result for our problem!

Weil cohomology theories

Most[¶] knowledge about $Z(X, T)$ comes from a construction of the sort predicted by Weil: construct finite-dimensional vector spaces $\{H^i(X)\}_{i=0}^{2d}$ over a field K of characteristic[‖] 0, each equipped with a linear transformation F, for which the **Lefschetz fixed point formula** holds:

$$\#X(\mathbb{F}_{q^n}) = \sum_{i=0}^{2d} (-1)^i \operatorname{Trace}(F^n, H^i(X)).$$

This immediately implies

$$Z(X, T) = \prod_{i=0}^{2d} \det(1 - TF, H^i(X))^{(-1)^{i+1}}.$$

[¶]There are a few recent algorithms that short-circuit this paradigm, especially in the context of computing Hasse–Weil L-functions over number fields.

[‖]If $\operatorname{char}(K) > 0$, the last equation becomes a congruence congruence modulo $\operatorname{char}(K)$, which doesn't uniquely determine $Z(X, T)$.

Weil cohomology theories

Most[¶] knowledge about $Z(X, T)$ comes from a construction of the sort predicted by Weil: construct finite-dimensional vector spaces $\{H^i(X)\}_{i=0}^{2d}$ over a field K of characteristic[‖] 0, each equipped with a linear transformation F, for which the **Lefschetz fixed point formula** holds:

$$\#X(\mathbb{F}_{q^n}) = \sum_{i=0}^{2d} (-1)^i \operatorname{Trace}(F^n, H^i(X)).$$

This immediately implies

$$Z(X, T) = \prod_{i=0}^{2d} \det(1 - TF, H^i(X))^{(-1)^{i+1}}.$$

[¶]There are a few recent algorithms that short-circuit this paradigm, especially in the context of computing Hasse–Weil L-functions over number fields.

[‖]If $\operatorname{char}(K) > 0$, the last equation becomes a congruence congruence modulo $\operatorname{char}(K)$, which doesn't uniquely determine $Z(X, T)$.

How to use a Weil cohomology

Unfortunately, there is no possible** Weil cohomology with $K = \mathbb{Q}$. In the known examples, K is a finite extension of $\mathbb{Q}_\ell$ for some prime ℓ (which may or may not be p).

This poses a challenge from the point of view of algorithms, as one cannot perform exact arithmetic in such K. Instead one uses careful approximations by analogy with computations in $\mathbb{R}$.

Fortunately, the Weil conjectures give enough control on the size of the coefficients of $Z(X, T)$ that a (rigorous) sufficiently accurate computation over K determines the answer uniquely.

For étale cohomology, instead of working with coefficients in a fixed $\mathbb{Q}_\ell$, we can consider $\mathbb{F}_\ell$-coefficients for multiple primes ℓ. These primes can be taken to be **logarithmic** in the other parameters; we then conclude using the Chinese remainder theorem.

**Serre deduces this from an analysis of supersingular elliptic curves.

How to use a Weil cohomology

Unfortunately, there is no possible** Weil cohomology with $K = \mathbb{Q}$. In the known examples, K is a finite extension of $\mathbb{Q}_\ell$ for some prime ℓ (which may or may not be p).

This poses a challenge from the point of view of algorithms, as one cannot perform exact arithmetic in such K. Instead one uses careful approximations by analogy with computations in $\mathbb{R}$.

Fortunately, the Weil conjectures give enough control on the size of the coefficients of $Z(X, T)$ that a (rigorous) sufficiently accurate computation over K determines the answer uniquely.

For étale cohomology, instead of working with coefficients in a fixed $\mathbb{Q}_\ell$, we can consider $\mathbb{F}_\ell$-coefficients for multiple primes ℓ. These primes can be taken to be **logarithmic** in the other parameters; we then conclude using the Chinese remainder theorem.

**Serre deduces this from an analysis of supersingular elliptic curves.

How to use a Weil cohomology

Unfortunately, there is no possible** Weil cohomology with $K = \mathbb{Q}$. In the known examples, K is a finite extension of $\mathbb{Q}_\ell$ for some prime ℓ (which may or may not be p).

This poses a challenge from the point of view of algorithms, as one cannot perform exact arithmetic in such K. Instead one uses careful approximations by analogy with computations in $\mathbb{R}$.

Fortunately, the Weil conjectures give enough control on the size of the coefficients of $Z(X, T)$ that a (rigorous) sufficiently accurate computation over K determines the answer uniquely.

For étale cohomology, instead of working with coefficients in a fixed $\mathbb{Q}_\ell$, we can consider $\mathbb{F}_\ell$-coefficients for multiple primes ℓ. These primes can be taken to be **logarithmic** in the other parameters; we then conclude using the Chinese remainder theorem.

**Serre deduces this from an analysis of supersingular elliptic curves.

How to use a Weil cohomology

Unfortunately, there is no possible** Weil cohomology with $K = \mathbb{Q}$. In the known examples, K is a finite extension of $\mathbb{Q}_\ell$ for some prime ℓ (which may or may not be p).

This poses a challenge from the point of view of algorithms, as one cannot perform exact arithmetic in such K. Instead one uses careful approximations by analogy with computations in $\mathbb{R}$.

Fortunately, the Weil conjectures give enough control on the size of the coefficients of $Z(X, T)$ that a (rigorous) sufficiently accurate computation over K determines the answer uniquely.

For étale cohomology, instead of working with coefficients in a fixed $\mathbb{Q}_\ell$, we can consider $\mathbb{F}_\ell$-coefficients for multiple primes ℓ. These primes can be taken to be **logarithmic** in the other parameters; we then conclude using the Chinese remainder theorem.

**Serre deduces this from an analysis of supersingular elliptic curves.

Contents

A general result

Theorem (Lauder–Wan, Harvey)

There is an algorithm to compute $Z(X, T)$ from X in time polynomial in p, $\deg X$, and $\log_p q$.

Note that p is not polynomial in $\log q$. While one can reduce the dependence on p to about $p^{1/2}$ (Harvey), these methods are **not** relevant[††] when $q = p$ is large (e.g., in many cryptography applications).

Such algorithms are based on Dwork's original proof[‡‡] that $Z(X, T)$ is rational. They do not seem to be practical in general, but for curves Kyng has recently achieved excellent results with Harvey's approach.

[††] This might be too pessimistic: Robert has a recent preprint computing canonical lifts of elliptic curves with polynomial dependence on $\log p$.

[‡‡] I consider this approach "cohomological" even through neither Dwork's original proof, nor the variant implicit in Harvey's work, directly constructs a cohomology theory; what is more visible is the analogue of the Lefschetz fixed point formula.

A general result

Theorem (Lauder–Wan, Harvey)

There is an algorithm to compute $Z(X, T)$ from X in time polynomial in p, $\deg X$, and $\log_p q$.

Note that p is not polynomial in $\log q$. While one can reduce the dependence on p to about $p^{1/2}$ (Harvey), these methods are **not** relevant[††] when $q = p$ is large (e.g., in many cryptography applications).

Such algorithms are based on Dwork's original proof[‡‡] that $Z(X, T)$ is rational. They do not seem to be practical in general, but for curves Kyng has recently achieved excellent results with Harvey's approach.

[††] This might be too pessimistic: Robert has a recent preprint computing canonical lifts of elliptic curves with polynomial dependence on $\log p$.

[‡‡] I consider this approach "cohomological" even through neither Dwork's original proof, nor the variant implicit in Harvey's work, directly constructs a cohomology theory; what is more visible is the analogue of the Lefschetz fixed point formula.

A very specific result

Theorem (K)

Assume $p \neq 2$ and let X be a **hyperelliptic** *curve of the form*

$$y^2 = P(x), \qquad \deg(P) = 2g + 1.$$

Then there is an algorithm to compute $Z(X, T)$ from X in time polynomial in p, g, and $\log_p q$.

Compared to the previous slide, the implied exponents/constants are much better, so this can be implemented at scale (see PARI, Magma, Sage).

This is based on a p-adic cohomology theory of Monsky–Washnitzer (for smooth affines), later generalized to **rigid cohomology** by Berthelot (closely related to **crystalline cohomology**).

A very specific result

Theorem (K)

Assume $p \neq 2$ and let X be a **hyperelliptic** *curve of the form*

$$y^2 = P(x), \qquad \deg(P) = 2g + 1.$$

Then there is an algorithm to compute $Z(X, T)$ from X in time polynomial in p, g, and $\log_p q$.

Compared to the previous slide, the implied exponents/constants are much better, so this can be implemented at scale (see PARI, Magma, Sage).

This is based on a p-adic cohomology theory of Monsky–Washnitzer (for smooth affines), later generalized to **rigid cohomology** by Berthelot (closely related to **crystalline cohomology**).

A very specific result

Theorem (K)

Assume $p \neq 2$ and let X be a **hyperelliptic** *curve of the form*

$$y^2 = P(x), \qquad \deg(P) = 2g + 1.$$

Then there is an algorithm to compute $Z(X, T)$ from X in time polynomial in p, g, and $\log_p q$.

Compared to the previous slide, the implied exponents/constants are much better, so this can be implemented at scale (see PARI, Magma, Sage).

This is based on a p-adic cohomology theory of Monsky–Washnitzer (for smooth affines), later generalized to **rigid cohomology** by Berthelot (closely related to **crystalline cohomology**).

Beyond the very specific result

Since the previous theorem, numerous generalizations have been developed. Two examples:

- Tuitman can handle arbitrary curves. This result plays an important role in some recent applications of the **Chabauty–Kim method** in Diophantine geometry (e.g., finding the rational points on modular curves of the form $X_{\mathrm{ns}}^{+}(N)$).
- Costa–Harvey–K can handle nondegenerate hypersurfaces in toric varieties (in practice up to dimension 4).

Deformations

One can also compute p-adic cohomology in one-parameter families (Lauder), using the Gauss–Manin connection to solve for the parametric Frobenius action.

This has been implemented at scale for some families of curves by Hubrechts, but not beyond this.

However, the key step of computing the Gauss–Manin connection is not at all p-adic! Some good ideas for this may come from elsewhere, e.g., mathematical physics (Movasati).

Deformations

One can also compute p-adic cohomology in one-parameter families (Lauder), using the Gauss–Manin connection to solve for the parametric Frobenius action.

This has been implemented at scale for some families of curves by Hubrechts, but not beyond this.

However, the key step of computing the Gauss–Manin connection is not at all p-adic! Some good ideas for this may come from elsewhere, e.g., mathematical physics (Movasati).

Deformations

One can also compute p-adic cohomology in one-parameter families (Lauder), using the Gauss–Manin connection to solve for the parametric Frobenius action.

This has been implemented at scale for some families of curves by Hubrechts, but not beyond this.

However, the key step of computing the Gauss–Manin connection is not at all p-adic! Some good ideas for this may come from elsewhere, e.g., mathematical physics (Movasati).

Contents

Schoof's algorithm

Theorem (Schoof)

For X an elliptic curve (e.g., $y^2 = x^3 + Ax + B$), there is an algorithm to compute $Z(X, T)$ from X in time polynomial in $\log q$.

Sketch of proof.

We need only the single integer $a := q + 1 - \#X(\mathbb{F}_q)$ for which $|a| \leq 2\sqrt{q}$. For each prime $\ell \neq p$, the ℓ-torsion points of X form a two-dimensional $\mathbb{F}_\ell$-vector space on which Frobenius acts with trace a. This exposes $a \pmod{\ell}$; use the Chinese remainder theorem to finish. □

Schoof's algorithm

Theorem (Schoof)

For X an elliptic curve (e.g., $y^2 = x^3 + Ax + B$), there is an algorithm to compute $Z(X, T)$ from X in time polynomial in $\log q$.

Sketch of proof.

We need only the single integer $a := q + 1 - \#X(\mathbb{F}_q)$ for which $|a| \leq 2\sqrt{q}$. For each prime $\ell \neq p$, the ℓ-torsion points of X form a two-dimensional $\mathbb{F}_\ell$-vector space on which Frobenius acts with trace a. This exposes $a \pmod{\ell}$; use the Chinese remainder theorem to finish. □

Pila's generalization of Schoof

Theorem (Pila)

For X a curve of **fixed** *genus g, there is an algorithm to compute $Z(X,T)$ from X in time polynomial in $\log q$.*

Sketch of proof.

Repeat Schoof's algorithm, but replacing X with its **Jacobian variety** J (since X itself no longer has a group structure). For each prime $\ell \neq p$, the ℓ-torsion points of J form a $2g$-dimensional $\mathbb{F}_\ell$-vector space on which Frob^n acts with trace $q^n + 1 - \#X(\mathbb{F}_{q^n})$. □

Sadly the dependence on g is (at least) exponential. One reason is that the ℓ-torsion points are used as a "black box group", on which we cannot efficiently read off the trace (but see below). Achieving polynomial dependence on both g and $\log q$ remains open.

Pila's generalization of Schoof

Theorem (Pila)

For X a curve of **fixed** *genus g, there is an algorithm to compute $Z(X,T)$ from X in time polynomial in $\log q$.*

Sketch of proof.

Repeat Schoof's algorithm, but replacing X with its **Jacobian variety** J (since X itself no longer has a group structure). For each prime $\ell \neq p$, the ℓ-torsion points of J form a $2g$-dimensional $\mathbb{F}_\ell$-vector space on which Frob^n acts with trace $q^n + 1 - \#X(\mathbb{F}_{q^n})$. □

Sadly the dependence on g is (at least) exponential. One reason is that the ℓ-torsion points are used as a "black box group", on which we cannot efficiently read off the trace (but see below). Achieving polynomial dependence on both g and $\log q$ remains open.

Pila's generalization of Schoof

Theorem (Pila)

For X a curve of **fixed** *genus g, there is an algorithm to compute $Z(X,T)$ from X in time polynomial in $\log q$.*

Sketch of proof.

Repeat Schoof's algorithm, but replacing X with its **Jacobian variety** J (since X itself no longer has a group structure). For each prime $\ell \neq p$, the ℓ-torsion points of J form a $2g$-dimensional $\mathbb{F}_\ell$-vector space on which Frob^n acts with trace $q^n + 1 - \#X(\mathbb{F}_{q^n})$. □

Sadly the dependence on g is (at least) exponential. One reason is that the ℓ-torsion points are used as a "black box group", on which we cannot efficiently read off the trace (but see below). Achieving polynomial dependence on both g and $\log q$ remains open.

But is this cohomology?

Yes, the ℓ-torsion points of J can also be interpreted as the first étale homology group of X with $\mathbb{F}_\ell$-coefficients.

Unfortunately, higher étale cohomology groups do not admit such direct geometric interpretation. For that matter, their definition is not itself intrinsically computable (it involves some highly infinite constructions like Grothendieck topologies).

Good news: Madore and Orgogozo have shown that higher étale cohomology groups are algorithmically computable.

Bad news: the proof includes **no** complexity analysis.

But is this cohomology?

Yes, the ℓ-torsion points of J can also be interpreted as the first étale homology group of X with $\mathbb{F}_\ell$-coefficients.

Unfortunately, higher étale cohomology groups do not admit such direct geometric interpretation. For that matter, their definition is not itself intrinsically computable (it involves some highly infinite constructions like Grothendieck topologies).

Good news: Madore and Orgogozo have shown that higher étale cohomology groups are algorithmically computable.

Bad news: the proof includes **no** complexity analysis.

But is this cohomology?

Yes, the ℓ-torsion points of J can also be interpreted as the first étale homology group of X with $\mathbb{F}_\ell$-coefficients.

Unfortunately, higher étale cohomology groups do not admit such direct geometric interpretation. For that matter, their definition is not itself intrinsically computable (it involves some highly infinite constructions like Grothendieck topologies).

Good news: Madore and Orgogozo have shown that higher étale cohomology groups are algorithmically computable.

Bad news: the proof includes **no** complexity analysis.

But is this cohomology?

Yes, the ℓ-torsion points of J can also be interpreted as the first étale homology group of X with $\mathbb{F}_\ell$-coefficients.

Unfortunately, higher étale cohomology groups do not admit such direct geometric interpretation. For that matter, their definition is not itself intrinsically computable (it involves some highly infinite constructions like Grothendieck topologies).

Good news: Madore and Orgogozo have shown that higher étale cohomology groups are algorithmically computable.

Bad news: the proof includes **no** complexity analysis.

Coming attractions

Very recently, Levrat has developed an algorithm to compute étale cohomology on curves with (locally constant torsion) coefficients, with a complexity analysis. This does not yet yield polynomial time algorithms, but this work is ongoing.

Better yet, it opens the door to inductive constructions in higher dimensions using Lefschetz pencils. This mirrors the theoretical use of pencils as our only tool to prove anything of substance about higher-degree étale cohomology!

Coming attractions

Very recently, Levrat has developed an algorithm to compute étale cohomology on curves with (locally constant torsion) coefficients, with a complexity analysis. This does not yet yield polynomial time algorithms, but this work is ongoing.

Better yet, it opens the door to inductive constructions in higher dimensions using Lefschetz pencils. This mirrors the theoretical use of pencils as our only tool to prove anything of substance about higher-degree étale cohomology!

Fourier coefficients of modular forms

Theorem (Couveignes–Edixhoven)

Let f be a fixed modular form of weight $k \geq 2$ with coefficients in some number field. Under GRH (for Dedekind L-functions), there exists an algorithm which, given a **factored** *positive integer N, computes the N-th Fourier coefficient of f in time polynomial in* $\log N$.

For example, this means that for p prime, $\tau(p)$ is computable in time polynomial in $\log p$.

The proof is **very** similar to Schoof–Pila: one computes the mod-ℓ Galois representation associated to f for $O(\log N)$ different primes ℓ.

I do not know how the complexity behaves as you vary f.

Fourier coefficients of modular forms

Theorem (Couveignes–Edixhoven)

Let f be a fixed modular form of weight $k \geq 2$ with coefficients in some number field. Under GRH (for Dedekind L-functions), there exists an algorithm which, given a **factored** *positive integer N, computes the N-th Fourier coefficient of f in time polynomial in* $\log N$.

For example, this means that for p prime, $\tau(p)$ is computable in time polynomial in $\log p$.

The proof is **very** similar to Schoof–Pila: one computes the mod-ℓ Galois representation associated to f for $O(\log N)$ different primes ℓ.

I do not know how the complexity behaves as you vary f.

Fourier coefficients of modular forms

Theorem (Couveignes–Edixhoven)

Let f be a fixed modular form of weight $k \geq 2$ with coefficients in some number field. Under GRH (for Dedekind L-functions), there exists an algorithm which, given a **factored** *positive integer N, computes the N-th Fourier coefficient of f in time polynomial in* $\log N$.

For example, this means that for p prime, $\tau(p)$ is computable in time polynomial in $\log p$.

The proof is **very** similar to Schoof–Pila: one computes the mod-ℓ Galois representation associated to f for $O(\log N)$ different primes ℓ.

I do not know how the complexity behaves as you vary f.

Fourier coefficients of modular forms

Theorem (Couveignes–Edixhoven)

Let f be a fixed modular form of weight $k \geq 2$ with coefficients in some number field. Under GRH (for Dedekind L-functions), there exists an algorithm which, given a **factored** *positive integer N, computes the N-th Fourier coefficient of f in time polynomial in* $\log N$.

For example, this means that for p prime, $\tau(p)$ is computable in time polynomial in $\log p$.

The proof is **very** similar to Schoof–Pila: one computes the mod-ℓ Galois representation associated to f for $O(\log N)$ different primes ℓ.

I do not know how the complexity behaves as you vary f.

Contents

Classical vs. quantum algorithms

In the hope (or fear) that quantum computing will soon be available at scale, one may also ask about complexity in this regime.

One key difference is **Shor's algorithm**, which gives polynomial time algorithms for things like:

- integer factorization;
- discrete logarithms in any **black box group**;
- in particular, finding **orders** of black box group;
- the **hidden subgroup problem** in abelian groups.

Classical vs. quantum algorithms

In the hope (or fear) that quantum computing will soon be available at scale, one may also ask about complexity in this regime.

One key difference is **Shor's algorithm**, which gives polynomial time algorithms for things like:

- integer factorization;
- discrete logarithms in any **black box group**;
- in particular, finding **orders** of black box group;
- the **hidden subgroup problem** in abelian groups.

Classical vs. quantum algorithms

In the hope (or fear) that quantum computing will soon be available at scale, one may also ask about complexity in this regime.

One key difference is **Shor's algorithm**, which gives polynomial time algorithms for things like:

- integer factorization;
- discrete logarithms in any **black box group**;
- in particular, finding **orders** of black box group;
- the **hidden subgroup problem** in abelian groups.

Classical vs. quantum algorithms

In the hope (or fear) that quantum computing will soon be available at scale, one may also ask about complexity in this regime.

One key difference is **Shor's algorithm**, which gives polynomial time algorithms for things like:

- integer factorization;
- discrete logarithms in any **black box group**;
- in particular, finding **orders** of black box group;
- the **hidden subgroup problem** in abelian groups.

Classical vs. quantum algorithms

In the hope (or fear) that quantum computing will soon be available at scale, one may also ask about complexity in this regime.

One key difference is **Shor's algorithm**, which gives polynomial time algorithms for things like:

- integer factorization;
- discrete logarithms in any **black box group**;
- in particular, finding **orders** of black box group;
- the **hidden subgroup problem** in abelian groups.

Classical vs. quantum algorithms

In the hope (or fear) that quantum computing will soon be available at scale, one may also ask about complexity in this regime.

One key difference is **Shor's algorithm**, which gives polynomial time algorithms for things like:

- integer factorization;
- discrete logarithms in any **black box group**;
- in particular, finding **orders** of black box group;
- the **hidden subgroup problem** in abelian groups.

A quantum complexity result

Theorem (K)

For X a curve of genus g, there is a quantum algorithm to compute $Z(X,T)$ in time polynomial in g and $\log q$. (This remains open for classical algorithms.)

This again uses the Jacobian variety J of X: if

$$Z(X,T) = \frac{P(T)}{(1-T)(1-qT)}, \text{ then } \#J(\mathbb{F}_{q^n}) = P(1)P(\zeta_n)\cdots P(\zeta_n^{n-1}).$$

We exhibit J as a black box group using divisor classes, use Shor to compute $\#J(\mathbb{F}_{q^n})$ for $n = 1, \ldots, O(g)$, and recover P from these by an elementary calculation.

Question: This generalizes immediately to abelian varieties. What about other classes of varieties (e.g., K3 surfaces)?

A quantum complexity result

Theorem (K)

For X a curve of genus g, there is a quantum algorithm to compute $Z(X,T)$ in time polynomial in g and $\log q$. (This remains open for classical algorithms.)

This again uses the Jacobian variety J of X: if

$$Z(X,T) = \frac{P(T)}{(1-T)(1-qT)}, \text{ then } \#J(\mathbb{F}_{q^n}) = P(1)P(\zeta_n)\cdots P(\zeta_n^{n-1}).$$

We exhibit J as a black box group using divisor classes, use Shor to compute $\#J(\mathbb{F}_{q^n})$ for $n = 1, \ldots, O(g)$, and recover P from these by an elementary calculation.

Question: This generalizes immediately to abelian varieties. What about other classes of varieties (e.g., K3 surfaces)?

A quantum complexity result

Theorem (K)

For X a curve of genus g, there is a quantum algorithm to compute $Z(X,T)$ in time polynomial in g and $\log q$. (This remains open for classical algorithms.)

This again uses the Jacobian variety J of X: if

$$Z(X,T) = \frac{P(T)}{(1-T)(1-qT)}, \text{ then } \#J(\mathbb{F}_{q^n}) = P(1)P(\zeta_n)\cdots P(\zeta_n^{n-1}).$$

We exhibit J as a black box group using divisor classes, use Shor to compute $\#J(\mathbb{F}_{q^n})$ for $n = 1, \ldots, O(g)$, and recover P from these by an elementary calculation.

Question: This generalizes immediately to abelian varieties. What about other classes of varieties (e.g., K3 surfaces)?

A quantum complexity result

Theorem (K)

For X a curve of genus g, there is a quantum algorithm to compute $Z(X,T)$ in time polynomial in g and $\log q$. (This remains open for classical algorithms.)

This again uses the Jacobian variety J of X: if

$$Z(X,T) = \frac{P(T)}{(1-T)(1-qT)}, \text{ then } \#J(\mathbb{F}_{q^n}) = P(1)P(\zeta_n)\cdots P(\zeta_n^{n-1}).$$

We exhibit J as a black box group using divisor classes, use Shor to compute $\#J(\mathbb{F}_{q^n})$ for $n = 1, \ldots, O(g)$, and recover P from these by an elementary calculation.

Question: This generalizes immediately to abelian varieties. What about other classes of varieties (e.g., K3 surfaces)?

Another approach

In quantum algorithms, Pila's algorithm is also polynomial in g and $\log q$. That is because the trace of a "black box linear transformation" can be recovered using Shor (as an instance of the hidden subgroup problem).

Question: Following Levrat, can one (efficiently) represent higher-degree étale cohomology groups as black box groups? If so, this would lead to a polynomial time quantum algorithm for computing $Z(X, T)$ for X of any dimension.

Another approach

In quantum algorithms, Pila's algorithm is also polynomial in g and $\log q$. That is because the trace of a "black box linear transformation" can be recovered using Shor (as an instance of the hidden subgroup problem).

Question: Following Levrat, can one (efficiently) represent higher-degree étale cohomology groups as black box groups? If so, this would lead to a polynomial time quantum algorithm for computing $Z(X, T)$ for X of any dimension.