A survey of $p$-adic approaches to zeta functions (plus a new approach)

Kiran S. Kedlaya

Department of Mathematics, University of California, San Diego
kedlaya@ucsd.edu
http://math.ucsd.edu/~kedlaya/slides/

GeoCrypt 2013: Geometry and Cryptography
Université de la Polynésie Française, Tahiti, October 11, 2013

The “new approach” material is joint work in progress with David Harvey (University of New South Wales) and Edgar Costa (New York University).

Supported by NSF (grant DMS-1101343), UCSD (Warschawski chair).
Contents

1. Generalities of zeta functions
2. The zeta function problem, and $p$-adic approaches
3. General algorithms
4. Algorithms for curves
5. Beyond hyperelliptic curves: nondegenerate toric hypersurfaces
6. Controlled reduction in $p$-adic cohomology
Generalities of zeta functions

The zeta function problem, and $p$-adic approaches

General algorithms

Algorithms for curves

Beyond hyperelliptic curves: nondegenerate toric hypersurfaces

Controlled reduction in $p$-adic cohomology
Let $X$ be a scheme of finite type over $\mathbb{Z}$. Its *zeta function* is the formal Dirichlet series

$$Z(X, s) = \prod_{x \in X^\circ} (1 - \#\kappa(x)^{-s})^{-1}$$

where $X^\circ$ is the set of closed points of $X$ and $\kappa(x)$ is the residue field of $x \in X^\circ$. For example, if $X = \text{Spec } \mathbb{Z}$, then $Z(X, s)$ is the Riemann zeta function

$$\zeta(s) = \prod_p (1 - p^{-s})^{-1} = \sum_{n=1}^{\infty} n^{-s}.$$ 

It is expected that $Z(X, s)$ has various nice properties, some of which are very deep open problems (analytic continuation, functional equation, analogue of the Riemann hypothesis, automorphicity, functoriality...).
The Weil conjectures

From now on, let $X$ be an algebraic variety over $\mathbb{F}_q$. The following facts are known (by Dwork, Grothendieck, Deligne, et al.):

- We have
  \[ Z(X, T) = \exp \left( \sum_{i=1}^{\infty} \frac{T^i}{i} \#X(\mathbb{F}_{q^i}) \right), \quad T = q^{-s}. \]

- $Z(X, T) \in \mathbb{Z}[[T]]$ represents a rational function of $T$.
- Assume from now on that $X$ is smooth proper of dimension $n$. Then
  \[ Z(X, T) = \prod_{i=0}^{2n} P_i(X, T)^{(-1)^{i+1}} = \frac{P_1(X, T) \cdots P_{2n-1}(X, T)}{P_0(X, T) \cdots P_{2n}(X, T)} \]
  for some $P_i(X, T) \in 1 + T\mathbb{Z}[T]$ with $\mathbb{C}$-roots of norm $q^{-i/2}$. Also,
  \[ P_{2n-i}(X, q^n T^{-1}) = \pm q^* T^* P_i(X, T). \]
- If $X$ lifts to characteristic 0, the $i$-th Betti number of the lift is $\deg P_i$. 

Kiran S. Kedlaya (UCSD)
The Weil conjectures

From now on, let $X$ be an algebraic variety over $\mathbb{F}_q$. The following facts are known (by Dwork, Grothendieck, Deligne, et al.):

- We have

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

- $Z(X, T) \in \mathbb{Z}[T]$ represents a rational function of $T$.
- Assume from now on that $X$ is smooth proper of dimension $n$. Then

$$Z(X, T) = \prod_{i=0}^{2n} P_i(X, T)^{(-1)^{i+1}} = \frac{P_1(X, T) \cdots P_{2n-1}(X, T)}{P_0(X, T) \cdots P_{2n}(X, T)}$$

for some $P_i(X, T) \in 1 + T \mathbb{Z}[T]$ with $\mathbb{C}$-roots of norm $q^{-i/2}$. Also,

$$P_{2n-i}(X, q^n T^{-1}) = \pm q^* T^* P_i(X, T).$$

- If $X$ lifts to characteristic 0, the $i$-th Betti number of the lift is $\deg P_i$. 
The Weil conjectures

From now on, let \( X \) be an algebraic variety over \( \mathbb{F}_q \). The following facts are known (by Dwork, Grothendieck, Deligne, et al.):

- We have
  \[
  Z(X, T) = \exp \left( \sum_{i=1}^{\infty} \frac{T^i}{i} \# X(\mathbb{F}_{q^i}) \right), \quad T = q^{-s}.
  \]

- \( Z(X, T) \in \mathbb{Z}[[T]] \) represents a rational function of \( T \).

Assume from now on that \( X \) is smooth proper of dimension \( n \). Then

\[
Z(X, T) = \prod_{i=0}^{2n} P_i(X, T)^{(-1)^{i+1}} = \frac{P_1(X, T) \cdots P_{2n-1}(X, T)}{P_0(X, T) \cdots P_{2n}(X, T)}
\]

for some \( P_i(X, T) \in 1 + T \mathbb{Z}[[T]] \) with \( \mathbb{C} \)-roots of norm \( q^{-i/2} \). Also,

\[
P_{2n-i}(X, q^n T^{-1}) = \pm q^* T^* P_i(X, T).
\]

- If \( X \) lifts to characteristic 0, the \( i \)-th Betti number of the lift is \( \deg P_i \).
The Weil conjectures

From now on, let $X$ be an algebraic variety over $\mathbb{F}_q$. The following facts are known (by Dwork, Grothendieck, Deligne, et al.):

- We have
  \[ Z(X, T) = \exp \left( \sum_{i=1}^{\infty} \frac{T^i}{i} \# X(\mathbb{F}_{q^i}) \right), \quad T = q^{-s}. \]

- $Z(X, T) \in \mathbb{Z}[T]$ represents a rational function of $T$.
- Assume from now on that $X$ is smooth proper of dimension $n$. Then
  \[ Z(X, T) = \prod_{i=0}^{2n} P_i(X, T)^{(-1)^{i+1}} = \frac{P_1(X, T) \cdots P_{2n-1}(X, T)}{P_0(X, T) \cdots P_{2n}(X, T)} \]
  for some $P_i(X, T) \in 1 + T\mathbb{Z}[T]$ with $\mathbb{C}$-roots of norm $q^{-i/2}$. Also,
  \[ P_{2n-i}(X, q^n T^{-1}) = \pm q^* T^* P_i(X, T). \]
- If $X$ lifts to characteristic 0, the $i$-th Betti number of the lift is $\deg P_i$. 
The Weil conjectures

From now on, let $X$ be an algebraic variety over $\mathbb{F}_q$. The following facts are known (by Dwork, Grothendieck, Deligne, et al.):

- We have
  \[
  Z(X, T) = \exp \left( \sum_{i=1}^{\infty} \frac{T^i}{i} \# X(\mathbb{F}_{q^i}) \right), \quad T = q^{-s}.
  \]

- $Z(X, T) \in \mathbb{Z}[T]$ represents a rational function of $T$.
- Assume from now on that $X$ is smooth proper of dimension $n$. Then
  \[
  Z(X, T) = \prod_{i=0}^{2n} P_i(X, T)^{(-1)^{i+1}} = \frac{P_1(X, T) \cdots P_{2n-1}(X, T)}{P_0(X, T) \cdots P_{2n}(X, T)}
  \]
  for some $P_i(X, T) \in 1 + T\mathbb{Z}[T]$ with $\mathbb{C}$-roots of norm $q^{-i/2}$. Also,
  \[
  P_{2n-i}(X, q^n T^{-1}) = \pm q^* T^* P_i(X, T).
  \]
- If $X$ lifts to characteristic 0, the $i$-th Betti number of the lift is $\deg P_i$. 

Kiran S. Kedlaya (UCSD)
If $X$ is a (smooth, projective, geometrically irreducible) curve of genus $g$ over $\mathbb{F}_q$, we have

$$Z(X, T) = \frac{P_1(X, T)}{(1 - T)(1 - qT)}$$

for some $P_1(X, T) \in 1 + T \mathbb{Z}[T]$ of degree $2g$ with $\mathbb{C}$-roots of norm $q^{-1/2}$ and satisfying

$$P_1(X, qT^{-1}) = q^g T^{-2g} P_1(X, T).$$

We may associate to $X$ its Jacobian abelian variety $J$, whose $\mathbb{F}_q$-points comprise the class (Picard) group of $X$. Then

$$P_i(J, T) = \bigwedge^i P_1(X, T) \quad (i = 0, \ldots, 2g);$$

in particular, $\#J(\mathbb{F}_q) = P_1(X, 1)$. This order is relevant if one wishes to use the discrete logarithm problem in $J(\mathbb{F}_q)$ for cryptography.
Zeta functions and Jacobians of curves

If $X$ is a (smooth, projective, geometrically irreducible) curve of genus $g$ over $\mathbb{F}_q$, we have

$$Z(X, T) = \frac{P_1(X, T)}{(1 - T)(1 - qT)}$$

for some $P_1(X, T) \in 1 + T \mathbb{Z}[T]$ of degree $2g$ with $\mathbb{C}$-roots of norm $q^{-1/2}$ and satisfying

$$P_1(X, qT^{-1}) = q^g T^{-2g} P_1(X, T).$$

We may associate to $X$ its Jacobian abelian variety $J$, whose $\mathbb{F}_q$-points comprise the class (Picard) group of $X$. Then

$$P_i(J, T) = \bigwedge^i P_1(X, T) \quad (i = 0, \ldots, 2g);$$

in particular, $\# J(\mathbb{F}_q) = P_1(X, 1)$. This order is relevant if one wishes to use the discrete logarithm problem in $J(\mathbb{F}_q)$ for cryptography.
The Lefschetz hyperplane theorem

In the examples we will consider, \( X \) will be not just proper but also projective. In this case, for \( H \) a hyperplane section,

\[ P_i(X, T) = P_i(H, T) \quad (i = 0, \ldots, n - 1). \]

In practice, this will mean that we need only compute \( P_n(X, T) \).
Contents

1 Generalities of zeta functions

2 The zeta function problem, and $p$-adic approaches

3 General algorithms

4 Algorithms for curves

5 Beyond hyperelliptic curves: nondegenerate toric hypersurfaces

6 Controlled reduction in $p$-adic cohomology
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...

Kiran S. Kedlaya (UCSD)
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...

Kiran S. Kedlaya (UCSD)
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...

Kiran S. Kedlaya (UCSD)
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...

Kiran S. Kedlaya (UCSD)
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...

Kiran S. Kedlaya (UCSD)
The zeta function problem

Given $X$ in an explicit form (i.e., defining equations), one would like an efficient algorithm to compute $Z(X, T)$. What would this tell us?

- Orders of Jacobians (e.g., for cryptography)
- Picard numbers (e.g., of K3 surfaces)
- Special values of $L$-functions
- Matching algebraic and automorphic $L$-functions (effective Langlands correspondence)
- Sato-Tate distributions
- Arithmetic aspects of mirror symmetry
- ...
Cohomology and zeta functions

Given $X$ in an explicit form, one would like to compute $Z(X, T)$. In principle, this is a finite computation once one bounds the degree of the rational function, but we want a computation which is feasible for interesting-sized examples, which the trivial computation (enumerating points over $\mathbb{F}_q, \mathbb{F}_{q^2}, \ldots$) usually is not!

Alternative: interpret $P_i(X, T)$ as the (reciprocal) characteristic polynomial of a linear transformation on étale cohomology. This only rarely yields a feasible computation, e.g., genus 1 curves (Schoof, Elkies, Atkin, et al.), genus 2 curves with real multiplication (Gaudry–Kohel–Smith), and (just barely) genus 2 curves (Gaudry–Schost).

Another alternative: use $p$-adic analogues of étale cohomology. These yield many more feasible computations, but generally only in small characteristic.
Cohomology and zeta functions

Given $X$ in an explicit form, one would like to compute $Z(X, T)$. In principle, this is a finite computation once one bounds the degree of the rational function, but we want a computation which is feasible for interesting-sized examples, which the trivial computation (enumerating points over $\mathbb{F}_q, \mathbb{F}_{q^2}, \ldots$) usually is not!

Alternative: interpret $P_i(X, T)$ as the (reciprocal) characteristic polynomial of a linear transformation on étale cohomology. This only rarely yields a feasible computation, e.g., genus 1 curves (Schoof, Elkies, Atkin, et al.), genus 2 curves with real multiplication (Gaudry–Kohel–Smith), and (just barely) genus 2 curves (Gaudry–Schost).

Another alternative: use $p$-adic analogues of étale cohomology. These yield many more feasible computations, but generally only in small characteristic.
Cohomology and zeta functions

Given $X$ in an explicit form, one would like to compute $Z(X, T)$. In principle, this is a finite computation once one bounds the degree of the rational function, but we want a computation which is feasible for interesting-sized examples, which the trivial computation (enumerating points over $\mathbb{F}_q, \mathbb{F}_{q^2}, \ldots$) usually is not!

Alternative: interpret $P_i(X, T)$ as the (reciprocal) characteristic polynomial of a linear transformation on étale cohomology. This only rarely yields a feasible computation, e.g., genus 1 curves (Schoof, Elkies, Atkin, et al.), genus 2 curves with real multiplication (Gaudry–Kohel–Smith), and (just barely) genus 2 curves (Gaudry–Schost).

Another alternative: use $p$-adic analogues of étale cohomology. These yield many more feasible computations, but generally only in small characteristic.
Sufficient $p$-adic precision

Write $q = p^a$ with $p$ prime.

Suppose $\deg P_i(X, T)$ is known for some $i$. Thanks to the bound on roots, for some *explicitly computable* value of $N$, knowledge of the coefficients of $P_i(X, T)$ modulo $p^N$ determines them exactly.

That is, we may compute $P_i(X, T)$ by computing it as a $p$-adic polynomial to sufficient precision, or by identifying it as the reciprocal characteristic polynomial of a $p$-adic matrix computed to sufficient precision.

In practice, even small reductions of the sufficient precision can translate into meaningful improvements in feasibility. Sample techniques:

- Work in terms of power sums of roots rather than elementary symmetric functions.
- Account for factors of $p$ in matrix entries via “Newton above Hodge”.
Sufficient $p$-adic precision

Write $q = p^a$ with $p$ prime.

Suppose $\deg P_i(X, T)$ is known for some $i$. Thanks to the bound on roots, for some explicitly computable value of $N$, knowledge of the coefficients of $P_i(X, T)$ modulo $p^N$ determines them exactly.

That is, we may compute $P_i(X, T)$ by computing it as a $p$-adic polynomial to sufficient precision, or by identifying it as the reciprocal characteristic polynomial of a $p$-adic matrix computed to sufficient precision.

In practice, even small reductions of the sufficient precision can translate into meaningful improvements in feasibility. Sample techniques:

- Work in terms of power sums of roots rather than elementary symmetric functions.
- Account for factors of $p$ in matrix entries via “Newton above Hodge”.
Sufficient $p$-adic precision

Write $q = p^a$ with $p$ prime.

Suppose $\deg P_i(X, T)$ is known for some $i$. Thanks to the bound on roots, for some explicitly computable value of $N$, knowledge of the coefficients of $P_i(X, T)$ modulo $p^N$ determines them exactly.

That is, we may compute $P_i(X, T)$ by computing it as a $p$-adic polynomial to sufficient precision, or by identifying it as the reciprocal characteristic polynomial of a $p$-adic matrix computed to sufficient precision.

In practice, even small reductions of the sufficient precision can translate into meaningful improvements in feasibility. Sample techniques:

- Work in terms of power sums of roots rather than elementary symmetric functions.
- Account for factors of $p$ in matrix entries via “Newton above Hodge”.

Sufficient $p$-adic precision

Write $q = p^a$ with $p$ prime.

Suppose $\deg P_i(X, T)$ is known for some $i$. Thanks to the bound on roots, for some explicitly computable value of $N$, knowledge of the coefficients of $P_i(X, T)$ modulo $p^N$ determines them exactly.

That is, we may compute $P_i(X, T)$ by computing it as a $p$-adic polynomial to sufficient precision, or by identifying it as the reciprocal characteristic polynomial of a $p$-adic matrix computed to sufficient precision.

In practice, even small reductions of the sufficient precision can translate into meaningful improvements in feasibility. Sample techniques:

- Work in terms of power sums of roots rather than elementary symmetric functions.
- Account for factors of $p$ in matrix entries via “Newton above Hodge”.
Contents

1 Generalities of zeta functions

2 The zeta function problem, and $p$-adic approaches

3 General algorithms

4 Algorithms for curves

5 Beyond hyperelliptic curves: nondegenerate toric hypersurfaces

6 Controlled reduction in $p$-adic cohomology
Dwork’s proof of rationality

Dwork’s proof of rationality of $Z(X, T)$ treats the case where $X$ is a hypersurface in affine space. This suffices by multiplicativity: if $U \subseteq X$ is open,

$$Z(X, T) = Z(U, T)Z(X - U, T).$$

To handle the special case, Dwork writes down a linear transformation on a $\mathbb{Q}_p$-vector space whose characteristic series computes $Z(X, T)$. This vector space is \textit{infinite dimensional} but the linear transformation is \textit{compact}, so it can be approximated by finite rank operators.

By explicitly approximating Dwork’s formula, Lauder and Wan obtained an algorithm for computing $Z(X, T)$. If $X$ is of degree $d$ and fixed dimension over $\mathbb{F}_q$ with $q = p^a$, this runs in time $\text{poly}(p, d, a)$.

Unfortunately, the implied exponents and constants seem to make this algorithm infeasible, even for curves. Harvey is working on a variant modeled on Hasse-Witt matrices, but this does not appear feasible either.
Dwork’s proof of rationality

Dwork’s proof of rationality of $Z(X, T)$ treats the case where $X$ is a hypersurface in affine space. This suffices by multiplicativity: if $U \subseteq X$ is open,

$$Z(X, T) = Z(U, T)Z(X - U, T).$$

To handle the special case, Dwork writes down a linear transformation on a $\mathbb{Q}_p$-vector space whose characteristic series computes $Z(X, T)$. This vector space is *infinite dimensional* but the linear transformation is *compact*, so it can be approximated by finite rank operators.

By explicitly approximating Dwork’s formula, Lauder and Wan obtained an algorithm for computing $Z(X, T)$. If $X$ is of degree $d$ and fixed dimension over $\mathbb{F}_q$ with $q = p^a$, this runs in time $\text{poly}(p, d, a)$.

Unfortunately, the implied exponents and constants seem to make this algorithm infeasible, even for curves. Harvey is working on a variant modeled on Hasse-Witt matrices, but this does not appear feasible either.
Dwork’s proof of rationality

Dwork’s proof of rationality of $Z(X, T)$ treats the case where $X$ is a hypersurface in affine space. This suffices by multiplicativity: if $U \subseteq X$ is open,

$$Z(X, T) = Z(U, T)Z(X - U, T).$$

To handle the special case, Dwork writes down a linear transformation on a $\mathbb{Q}_p$-vector space whose characteristic series computes $Z(X, T)$. This vector space is *infinite dimensional* but the linear transformation is *compact*, so it can be approximated by finite rank operators.

By explicitly approximating Dwork’s formula, Lauder and Wan obtained an algorithm for computing $Z(X, T)$. If $X$ is of degree $d$ and fixed dimension over $\mathbb{F}_q$ with $q = p^a$, this runs in time $\text{poly}(p, d, a)$.

Unfortunately, the implied exponents and constants seem to make this algorithm infeasible, even for curves. Harvey is working on a variant modeled on Hasse-Witt matrices, but this does not appear feasible either.
Contents

1 Generalities of zeta functions

2 The zeta function problem, and $p$-adic approaches

3 General algorithms

4 Algorithms for curves

5 Beyond hyperelliptic curves: nondegenerate toric hypersurfaces

6 Controlled reduction in $p$-adic cohomology
Let $X$ be an elliptic curve over $\mathbb{F}_q$ with $q = p^a$. If $X$ is supersingular then there are few possibilities for $Z(X, T)$ and they are easily distinguishable.

If $X$ is ordinary, it admits a unique lift $\tilde{X}$ to $\mathbb{Q}_q$ (the unramified extension of $\mathbb{Q}_p$ with residue field $\mathbb{F}_q$) with the same endomorphisms. Satoh (in characteristic $> 2$) and Fouquet–Gaudry–Harley (in characteristic 2) described a quadratically convergent iteration to compute $\tilde{X}$ and the lift of absolute Frobenius. One can then read off $Z(X, T)$ as the characteristic polynomial of the $q$-power Frobenius.

For fixed $p$, this algorithm is cubic in $a$ (whereas Schoof–Elkies–Atkin is quartic in $a$). This has been executed for $q = 2^{8009}$ (FGH, 2000).

Unfortunately, this does not generalize well to hyperelliptic curves: the canonical lifting of the Jacobian exists but may not itself be a Jacobian. (Genus 2 might still be doable.)
Let $X$ be an elliptic curve over $\mathbb{F}_q$ with $q = p^a$. If $X$ is supersingular then there are few possibilities for $Z(X, T)$ and they are easily distinguishable.

If $X$ is ordinary, it admits a unique lift $\tilde{X}$ to $\mathbb{Q}_q$ (the unramified extension of $\mathbb{Q}_p$ with residue field $\mathbb{F}_q$) with the same endomorphisms. Satoh (in characteristic $> 2$) and Fouquet–Gaudry–Harley (in characteristic 2) described a quadratically convergent iteration to compute $\tilde{X}$ and the lift of absolute Frobenius. One can then read off $Z(X, T)$ as the characteristic polynomial of the $q$-power Frobenius.

For fixed $p$, this algorithm is cubic in $a$ (whereas Schoof–Elkies–Atkin is quartic in $a$). This has been executed for $q = 2^{8009}$ (FGH, 2000).

Unfortunately, this does not generalize well to hyperelliptic curves: the canonical lifting of the Jacobian exists but may not itself be a Jacobian. (Genus 2 might still be doable.)
Canonical liftings

Let $X$ be an elliptic curve over $\mathbb{F}_q$ with $q = p^a$. If $X$ is supersingular then there are few possibilities for $Z(X, T)$ and they are easily distinguishable.

If $X$ is ordinary, it admits a unique lift $\tilde{X}$ to $\mathbb{Q}_q$ (the unramified extension of $\mathbb{Q}_p$ with residue field $\mathbb{F}_q$) with the same endomorphisms. Satoh (in characteristic $> 2$) and Fouquet–Gaudry–Harley (in characteristic 2) described a quadratically convergent iteration to compute $\tilde{X}$ and the lift of absolute Frobenius. One can then read off $Z(X, T)$ as the characteristic polynomial of the $q$-power Frobenius.

For fixed $p$, this algorithm is cubic in $a$ (whereas Schoof–Elkies–Atkin is quartic in $a$). This has been executed for $q = 2^{8009}$ (FGH, 2000).

Unfortunately, this does not generalize well to hyperelliptic curves: the canonical lifting of the Jacobian exists but may not itself be a Jacobian. (Genus 2 might still be doable.)
Let $X$ be an elliptic curve over $\mathbb{F}_q$ with $q = p^a$. If $X$ is supersingular then there are few possibilities for $Z(X, T)$ and they are easily distinguishable.

If $X$ is ordinary, it admits a unique lift $\tilde{X}$ to $\mathbb{Q}_q$ (the unramified extension of $\mathbb{Q}_p$ with residue field $\mathbb{F}_q$) with the same endomorphisms. Satoh (in characteristic $>2$) and Fouquet–Gaudry–Harley (in characteristic 2) described a quadratically convergent iteration to compute $\tilde{X}$ and the lift of absolute Frobenius. One can then read off $Z(X, T)$ as the characteristic polynomial of the $q$-power Frobenius.

For fixed $p$, this algorithm is cubic in $a$ (whereas Schoof–Elkies–Atkin is quartic in $a$). This has been executed for $q = 2^{8009}$ (FGH, 2000).

Unfortunately, this does not generalize well to hyperelliptic curves: the canonical lifting of the Jacobian exists but may not itself be a Jacobian. (Genus 2 might still be doable.)
Mestre’s AGM iteration

Over $\mathbb{R}$, the arithmetic-geometric mean iteration

$$(x, y) \mapsto \left( \frac{x + y}{2}, \sqrt{xy} \right)$$

converges and computes elliptic integrals.

Let $X$ be an ordinary elliptic curve over $\mathbb{F}_q$ with $q = 2^a$. Using a 2-adic analogue of the AGM iteration, Mestre obtains a quadratically convergent iteration computing the *unit root* of the characteristic polynomial of Frobenius.

This algorithm is quadratic in $a$ and runs very well in practice. There are variants for $p > 2$ (Kohel) but these are less tested in practice.

A similar algorithm exists for ordinary hyperelliptic curves. There is exponential dependence on $g$, but $g = 2, 3$ are (probably) feasible.
Mestre’s AGM iteration

Over $\mathbb{R}$, the arithmetic-geometric mean mean iteration

$$(x, y) \mapsto \left(\frac{x + y}{2}, \sqrt{xy}\right)$$

converges and computes elliptic integrals.

Let $X$ be an ordinary elliptic curve over $\mathbb{F}_q$ with $q = 2^a$. Using a 2-adic analogue of the AGM iteration, Mestre obtains a quadratically convergent iteration computing the unit root of the characteristic polynomial of Frobenius.

This algorithm is quadratic in $a$ and runs very well in practice. There are variants for $p > 2$ (Kohel) but these are less tested in practice.

A similar algorithm exists for ordinary hyperelliptic curves. There is exponential dependence on $g$, but $g = 2, 3$ are (probably) feasible.
Mestre’s AGM iteration

Over $\mathbb{R}$, the arithmetic-geometric mean iteration

$$(x, y) \mapsto \left(\frac{x + y}{2}, \sqrt{xy}\right)$$

converges and computes elliptic integrals.

Let $X$ be an ordinary elliptic curve over $\mathbb{F}_q$ with $q = 2^a$. Using a 2-adic analogue of the AGM iteration, Mestre obtains a quadratically convergent iteration computing the unit root of the characteristic polynomial of Frobenius.

This algorithm is quadratic in $a$ and runs very well in practice. There are variants for $p > 2$ (Kohel) but these are less tested in practice.

A similar algorithm exists for ordinary hyperelliptic curves. There is exponential dependence on $g$, but $g = 2, 3$ are (probably) feasible.
Mestre’s AGM iteration

Over \( \mathbb{R} \), the arithmetic-geometric mean iteration

\[
(x, y) \mapsto \left( \frac{x + y}{2}, \sqrt{xy} \right)
\]

converges and computes elliptic integrals.

Let \( X \) be an ordinary elliptic curve over \( \mathbb{F}_q \) with \( q = 2^a \). Using a 2-adic analogue of the AGM iteration, Mestre obtains a quadratically convergent iteration computing the unit root of the characteristic polynomial of Frobenius.

This algorithm is quadratic in \( a \) and runs very well in practice. There are variants for \( p > 2 \) (Kohel) but these are less tested in practice.

A similar algorithm exists for ordinary hyperelliptic curves. There is exponential dependence on \( g \), but \( g = 2, 3 \) are (probably) feasible.
For $X$ a hyperelliptic curve of genus $g$ (with $p > 2$ and having a rational Weierstrass point), Kedlaya (2001) described an algorithm for computing $Z(X, T)$ by realizing $P_1(X, T)$ as the characteristic polynomial of Frobenius on Monsky-Washnitzer cohomology of the affine curve obtained by removing the Weierstrass points. This runs in time $(pg^4a^3)^{1+\epsilon}$ and is feasible, both for small $p$ and large $g$ (e.g., $p = 3, g = 100$) and for large $p$ and small $g$ (e.g., $p \sim 2^{16}, g = 2$).

One can remove the restrictions on $p$ (Denef-Vercauteren) and the rational Weierstrass point (Harrison).

Harvey described a variant of this algorithm that improves the dependence on $p$ to $p^{1/2+\epsilon}$ (with some worsening in the other exponents). This is feasible for even larger $p$ and small $g$ (e.g., $p \sim 2^{32}, g = 3$).
For $X$ a hyperelliptic curve of genus $g$ (with $p > 2$ and having a rational Weierstrass point), Kedlaya (2001) described an algorithm for computing $Z(X, T)$ by realizing $P_1(X, T)$ as the characteristic polynomial of Frobenius on Monsky-Washnitzer cohomology of the affine curve obtained by removing the Weierstrass points. This runs in time $(pg^4a^3)^{1+\epsilon}$ and is feasible, both for small $p$ and large $g$ (e.g., $p = 3, g = 100$) and for large $p$ and small $g$ (e.g., $p \sim 2^{16}, g = 2$).

One can remove the restrictions on $p$ (Denef-Vercauteren) and the rational Weierstrass point (Harrison).

Harvey described a variant of this algorithm that improves the dependence on $p$ to $p^{1/2+\epsilon}$ (with some worsening in the other exponents). This is feasible for even larger $p$ and small $g$ (e.g., $p \sim 2^{32}, g = 3$).
Kedlaya’s cohomological method and Harvey’s variant

For \( X \) a hyperelliptic curve of genus \( g \) (with \( p > 2 \) and having a rational Weierstrass point), Kedlaya (2001) described an algorithm for computing \( Z(X, T) \) by realizing \( P_1(X, T) \) as the characteristic polynomial of Frobenius on Monsky-Washnitzer cohomology of the affine curve obtained by removing the Weierstrass points. This runs in time \((pg^4a^3)^{1+\epsilon}\) and is feasible, both for small \( p \) and large \( g \) (e.g., \( p = 3, g = 100 \)) and for large \( p \) and small \( g \) (e.g., \( p \sim 2^{16}, g = 2 \)).

One can remove the restrictions on \( p \) (Denef-Vercauteren) and the rational Weierstrass point (Harrison).

Harvey described a variant of this algorithm that improves the dependence on \( p \) to \( p^{1/2+\epsilon} \) (with some worsening in the other exponents). This is feasible for even larger \( p \) and small \( g \) (e.g., \( p \sim 2^{32}, g = 3 \)).
Deformation methods

Lauder proposed a general class of $p$-adic cohomological algorithms which involve the computation of a Frobenius action not on a single cohomology, but on a family of cohomology groups coming from a one-parameter family of varieties.

This was used by Hubrechts to reduce the space complexity of computing $Z(X, T)$ for hyperelliptic curves.
Lauder proposed a general class of $p$-adic cohomological algorithms which involve the computation of a Frobenius action not on a single cohomology, but on a family of cohomology groups coming from a one-parameter family of varieties.

This was used by Hubrechts to reduce the space complexity of computing $Z(X, T)$ for hyperelliptic curves.
Extensions of Kedlaya’s algorithm

The computation of Frobenius actions on Monsky-Washnitzer cohomology can be generalized to superelliptic curves (Gaudry–Gürel), $C_{a,b}$-curves (Denef–Vercauteren), nondegenerate curves (Castryck–Denef–Vercauteren), all curves (Tuitman).

An alternate approach, which may be more practical in the general case, uses the cup product duality (Besser–de Jeu–Escriva).
For a hyperelliptic curve over $\mathbb{Q}$, Harvey described a method for amortizing the computation of zeta functions over $\mathbb{F}_p$ for all $p \leq x$, to get average polynomial time (i.e., time $\text{poly}(\log(p), a, g)$ per prime). This incorporates an idea of Gerbicz from the context of computing Wilson quotients (i.e., $(p - 1)! \mod p^2$) using balanced remainder trees.

This has been implemented by Harvey and Sutherland in genus 1, 2 and outperforms all existing implementations (Pari, Magma, Smalljac).

This technique should apply to many other cohomological methods, e.g., deformation methods (Harvey–Tuitman).
Some improvements for hyperelliptic curves

For a hyperelliptic curve over $\mathbb{Q}$, Harvey described a method for amortizing the computation of zeta functions over $\mathbb{F}_p$ for all $p \leq x$, to get average polynomial time (i.e., time poly(log($p$), $a$, $g$) per prime). This incorporates an idea of Gerbicz from the context of computing Wilson quotients (i.e., $(p - 1)! \mod p^2$) using balanced remainder trees.

This has been implemented by Harvey and Sutherland in genus 1, 2 and outperforms all existing implementations (Pari, Magma, Smalljac).

This technique should apply to many other cohomological methods, e.g., deformation methods (Harvey–Tuitman).
Some improvements for hyperelliptic curves

For a hyperelliptic curve over $\mathbb{Q}$, Harvey described a method for amortizing the computation of zeta functions over $\mathbb{F}_p$ for all $p \leq x$, to get average polynomial time (i.e., time $\text{poly}(\log(p), a, g)$ per prime). This incorporates an idea of Gerbicz from the context of computing Wilson quotients (i.e., $(p - 1)! \mod p^2$) using balanced remainder trees.

This has been implemented by Harvey and Sutherland in genus 1, 2 and outperforms all existing implementations (Pari, Magma, Smalljac).

This technique should apply to many other cohomological methods, e.g., deformation methods (Harvey–Tuitman).
Contents

1 Generalities of zeta functions

2 The zeta function problem, and $p$-adic approaches

3 General algorithms

4 Algorithms for curves

5 Beyond hyperelliptic curves: nondegenerate toric hypersurfaces

6 Controlled reduction in $p$-adic cohomology
Beyond hyperelliptic curves: nondegenerate toric hypersurfaces

Projective hypersurfaces

For smooth projective hypersurfaces, Abbott–Kedlaya–Roe described an algorithm for computing \( Z(X, T) \) by working in the affine complement; we will see this trick again later. Unfortunately, the dependence on \( p \) goes like \( p^n \) for \( n = \dim(X) \). The analogue of Castryck–Denef–Vercauteren behaves similarly.

Some alternatives that alleviate the dependence on \( p \) are Lauder’s deformation method and fibration method. However, these seem to be feasible (so far) only for sparse polynomials (Pancratz–Tuitman).

Also available (and maybe feasible?) for sparse polynomials is Sperber–Voight, based on Dwork cohomology.

Hereafter, we describe a variant of AKR which has good (namely linear) dependence on \( p \), can handle dense polynomials, and is feasible (shown by example!). One tradeoff is that we restrict the class of projective hypersurfaces slightly, but as a bonus we pick up many more examples.
Beyond hyperelliptic curves: nondegenerate toric hypersurfaces

Projective hypersurfaces

For smooth projective hypersurfaces, Abbott–Kedlaya–Roe described an algorithm for computing $Z(X, T)$ by working in the affine complement; we will see this trick again later. Unfortunately, the dependence on $p$ goes like $p^n$ for $n = \dim(X)$. The analogue of Castryck–Denef–Vercauteren behaves similarly.

Some alternatives that alleviate the dependence on $p$ are Lauder’s deformation method and fibration method. However, these seem to be feasible (so far) only for sparse polynomials (Pancratz–Tuitman).

Also available (and maybe feasible?) for sparse polynomials is Sperber–Voight, based on Dwork cohomology.

Hereafter, we describe a variant of AKR which has good (namely linear) dependence on $p$, can handle dense polynomials, and is feasible (shown by example!). One tradeoff is that we restrict the class of projective hypersurfaces slightly, but as a bonus we pick up many more examples.
Projective hypersurfaces

For smooth projective hypersurfaces, Abbott–Kedlaya–Roe described an algorithm for computing $Z(X, T)$ by working in the affine complement; we will see this trick again later. Unfortunately, the dependence on $p$ goes like $p^n$ for $n = \dim(X)$. The analogue of Castryck–Denef–Vercauteren behaves similarly.

Some alternatives that alleviate the dependence on $p$ are Lauder’s deformation method and fibration method. However, these seem to be feasible (so far) only for sparse polynomials (Pancratz–Tuitman).

Also available (and maybe feasible?) for sparse polynomials is Sperber–Voight, based on Dwork cohomology.

Hereafter, we describe a variant of AKR which has good (namely linear) dependence on $p$, can handle dense polynomials, and is feasible (shown by example!). One tradeoff is that we restrict the class of projective hypersurfaces slightly, but as a bonus we pick up many more examples.
Projective hypersurfaces

For smooth projective hypersurfaces, Abbott–Kedlaya–Roe described an algorithm for computing $Z(X, T)$ by working in the affine complement; we will see this trick again later. Unfortunately, the dependence on $p$ goes like $p^n$ for $n = \dim(X)$. The analogue of Castryck–Denef–Vercauteren behaves similarly.

Some alternatives that alleviate the dependence on $p$ are Lauder’s deformation method and fibration method. However, these seem to be feasible (so far) only for sparse polynomials (Pancratz–Tuitman).

Also available (and maybe feasible?) for sparse polynomials is Sperber–Voight, based on Dwork cohomology.

Hereafter, we describe a variant of AKR which has good (namely linear) dependence on $p$, can handle dense polynomials, and is feasible (shown by example!). One tradeoff is that we restrict the class of projective hypersurfaces slightly, but as a bonus we pick up many more examples.
Lattices and differentials

Let $R$ be a ring. Let $L$ be a lattice of rank $n$. Let $L^\vee := \text{Hom}_\mathbb{Z}(L, \mathbb{Z})$ denote the dual lattice.

Let $R[L]$ denote the monoid algebra. Concretely, if we fix a basis $e_1, \ldots, e_n$ of $L$, we obtain an isomorphism

$$R[L] \cong R[x_1^{\pm}, \ldots, x_n^{\pm}], \quad [e_i] \mapsto x_i.$$

Each $\lambda \in L^\vee$ defines a derivation $\partial_\lambda$ on $R[L]$ via the formula

$$\partial_\lambda([v]) = \lambda(v)[v] \quad (v \in L);$$

these satisfy $\partial_{\lambda_1 + \lambda_2} = \partial_{\lambda_1} + \partial_{\lambda_2}$. With a basis as above, for $e_1^\vee, \ldots, e_n^\vee \in L^\vee$ the dual basis,

$$\partial_{e_i^\vee} = x_i \frac{\partial}{\partial x_i}.$$
Lattices and differentials

Let $R$ be a ring. Let $L$ be a lattice of rank $n$. Let $L^\vee := \text{Hom}_\mathbb{Z}(L, \mathbb{Z})$ denote the dual lattice.

Let $R[L]$ denote the monoid algebra. Concretely, if we fix a basis $e_1, \ldots, e_n$ of $L$, we obtain an isomorphism

$$R[L] \cong R[x_1^\pm, \ldots, x_n^\pm], \quad [e_i] \mapsto x_i.$$

Each $\lambda \in L^\vee$ defines a derivation $\partial_\lambda$ on $R[L]$ via the formula

$$\partial_\lambda([v]) = \lambda(v)[v] \quad (v \in L);$$

these satisfy $\partial_{\lambda_1 + \lambda_2} = \partial_{\lambda_1} + \partial_{\lambda_2}$. With a basis as above, for $e_1^\vee, \ldots, e_n^\vee \in L^\vee$ the dual basis,

$$\partial_{e_i^\vee} = x_i \frac{\partial}{\partial x_i}.$$
Lattices and differentials

Let $R$ be a ring. Let $L$ be a lattice of rank $n$. Let $L^\vee := \text{Hom}_\mathbb{Z}(L, \mathbb{Z})$ denote the dual lattice.

Let $R[L]$ denote the monoid algebra. Concretely, if we fix a basis $e_1, \ldots, e_n$ of $L$, we obtain an isomorphism

$$R[L] \cong R[x_1^\pm, \ldots, x_n^\pm], \quad [e_i] \mapsto x_i.$$  

Each $\lambda \in L^\vee$ defines a derivation $\partial_\lambda$ on $R[L]$ via the formula

$$\partial_\lambda([v]) = \lambda(v)[v] \quad (v \in L);$$

these satisfy $\partial_{\lambda_1 + \lambda_2} = \partial_{\lambda_1} + \partial_{\lambda_2}$. With a basis as above, for $e_1^\vee, \ldots, e_n^\vee \in L^\vee$ the dual basis,

$$\partial_{e_i^\vee} = x_i \frac{\partial}{\partial x_i}.$$
Let $\Delta$ be a convex lattice polytope of full dimension in $L_\mathbb{R} := L \otimes_\mathbb{Z} \mathbb{R}$, i.e., the convex hull of a finite subset of $L$ not contained in any hyperplane. The cone over this polytope is then a fan defining a (polarized) projective toric variety over $R$. In simple cases, this can be computed as

$$X := \text{Proj } P, \quad P := \bigoplus_{d=0}^{\infty} P_d, \quad P_d := R[d\Delta \cap L]$$

but in bad cases (e.g., for $\Delta = \text{Conv}(0, e_1, e_2, e_1 + e_2 + 3e_3)$) one must take $P$ to be Cox's *homogeneous coordinate ring*. For example, for $\Delta$ the simplex with vertices $0, e_1, \ldots, e_n$, we get projective space with its usual $\mathcal{O}(1)$. We similarly get weighted projective spaces, products, toric blowups, etc. Replacing $\Delta$ by $d\Delta$ preserves $X$ but replaces the polarization by its $d$-th power.
Polytopes and projective toric varieties

Let $\Delta$ be a convex lattice polytope of full dimension in $L_R := L \otimes \mathbb{Z} \mathbb{R}$, i.e., the convex hull of a finite subset of $L$ not contained in any hyperplane. The cone over this polytope is then a fan defining a (polarized) projective toric variety over $R$. In simple cases, this can be computed as

$$X := \text{Proj } P, \quad P := \bigoplus_{d=0}^{\infty} P_d, \quad P_d := R[d\Delta \cap L]$$

but in bad cases (e.g., for $\Delta = \text{Conv}(0, e_1, e_2, e_1 + e_2 + 3e_3)$) one must take $P$ to be Cox’s homogeneous coordinate ring.

For example, for $\Delta$ the simplex with vertices $0, e_1, \ldots, e_n$, we get projective space with its usual $\mathcal{O}(1)$. We similarly get weighted projective spaces, products, toric blowups, etc. Replacing $\Delta$ by $d\Delta$ preserves $X$ but replaces the polarization by its $d$-th power.
Nondegeneracy

We say \( f \in P_d \) is nondegenerate if the hypersurface

\[
Z_f := \text{Proj } P/(f)
\]

cut out by \( f \) has transversal intersection with each torus in the natural stratification of \( X \). In particular, this is required for the zero-dimensional strata, so \( f \) must have Newton polytope \( d\Delta \).

It is equivalent to require that the toric Jacobian ideal

\[
I_f = (f, \delta_\lambda(f) : \lambda \in L^\vee)
\]

is irrelevant, that is, the toric Jacobian ring \( J_f := P/I_f \) is module-finite over \( R \). This condition is generic for “nice” \( P \).

Note: if \( f \) is nondegenerate, then \( Z_f \) is “no more singular than \( X \)”. 
Nondegeneracy

We say \( f \in P_d \) is nondegenerate if the hypersurface

\[ Z_f := \text{Proj} \ P/(f) \]

cut out by \( f \) has transversal intersection with each torus in the natural stratification of \( X \). In particular, this is required for the zero-dimensional strata, so \( f \) must have Newton polytope \( d\Delta \).

It is equivalent to require that the toric Jacobian ideal

\[ I_f = (f, \delta_\lambda(f) : \lambda \in L^\vee) \]

is irrelevant, that is, the toric Jacobian ring \( J_f := P/I_f \) is module-finite over \( R \). This condition is generic for “nice” \( P \).

Note: if \( f \) is nondegenerate, then \( Z_f \) is “no more singular than \( X \).”
We say \( f \in P_d \) is nondegenerate if the hypersurface
\[
Z_f := \text{Proj } P/(f)
\]
cut out by \( f \) has transversal intersection with each torus in the natural stratification of \( X \). In particular, this is required for the zero-dimensional strata, so \( f \) must have Newton polytope \( d\Delta \).

It is equivalent to require that the toric Jacobian ideal
\[
I_f = (f, \delta_\lambda(f) : \lambda \in L^\vee)
\]
is irrelevant, that is, the toric Jacobian ring \( J_f := P/I_f \) is module-finite over \( R \). This condition is generic for “nice” \( P \).

Note: if \( f \) is nondegenerate, then \( Z_f \) is “no more singular than \( X \)”. 

Some examples of nondegenerate hypersurfaces

<table>
<thead>
<tr>
<th>$n$</th>
<th>Vertices of $\Delta$</th>
<th>Resulting hypersurface</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$0, de_1, de_2$</td>
<td>Smooth plane curve of genus $\left(\frac{d-1}{2}\right)$</td>
</tr>
<tr>
<td>2</td>
<td>$0, (2g + 1)e_1, e_2$</td>
<td>Odd hyperelliptic curve of genus $g$</td>
</tr>
<tr>
<td>2</td>
<td>$0, ae_1, be_2$</td>
<td>$C_{a,b}$-curve</td>
</tr>
<tr>
<td>2</td>
<td>$0, (g + 1)e_1, 2e_2, (g + 1)e_1 + 2e_2$</td>
<td>Even hyperelliptic curve of genus $g$</td>
</tr>
<tr>
<td>3</td>
<td>$0, 4e_1, 4e_2, 4e_3$</td>
<td>Quartic K3 surface</td>
</tr>
<tr>
<td>4</td>
<td>$0, 5e_1, \ldots, 5e_5$</td>
<td>Quintic Calabi-Yau threefold</td>
</tr>
</tbody>
</table>
Contents

1. Generalities of zeta functions
2. The zeta function problem, and $p$-adic approaches
3. General algorithms
4. Algorithms for curves
5. Beyond hyperelliptic curves: nondegenerate toric hypersurfaces
6. Controlled reduction in $p$-adic cohomology
Monsky-Washnitzer cohomology

From now on, work over $R = \mathbb{Z}_q$ and take $f \in P_1$ nondegenerate. (If $f \in P_d$ for $d > 0$, we may replace $\Delta$ with $d\Delta$ and then proceed.) Put $U_f := X \setminus Z_f$; this is an affine scheme with coordinate ring

$$S = \bigoplus_{m=0}^{\infty} f^{-m}P_m.$$ 

The weak completion $S^\dagger$ of $S$ consists of infinite series $\sum_{m=0}^{\infty} g_m f^{-m}$ with $g_m \in P_m$ such that for some $a, b > 0$ (depending on the series),

$$v_p(g_m) \geq am - b \quad (m \geq 0).$$

The Monsky-Washnitzer cohomology of $U_{\mathbb{F}_q}$ is the cohomology of the (continuous) de Rham complex $\Omega_{S^\dagger[p^{-1}]/\mathbb{Q}_q}$. 
Monsky-Washnitzer cohomology

From now on, work over $R = \mathbb{Z}_q$ and take $f \in P_1$ nondegenerate. (If $f \in P_d$ for $d > 0$, we may replace $\Delta$ with $d\Delta$ and then proceed.) Put $U_f := X \setminus Z_f$; this is an affine scheme with coordinate ring

$$S = \bigoplus_{m=0}^{\infty} f^{-m}P_m.$$ 

The weak completion $S^\dagger$ of $S$ consists of infinite series $\sum_{m=0}^{\infty} g_m f^{-m}$ with $g_m \in P_m$ such that for some $a, b > 0$ (depending on the series),

$$v_p(g_m) \geq am - b \quad (m \geq 0).$$

The Monsky-Washnitzer cohomology of $U_{\mathbb{F}_q}$ is the cohomology of the (continuous) de Rham complex $\Omega S^\dagger[p^{-1}]/\mathbb{Q}_q$. 
Action of Frobenius

Define a (semilinear) endomorphism $\sigma$ of $S^\dagger$ as the absolute Frobenius lift on $R$, the substitution $[v] \mapsto [v]^p$ on monomials, and

$$g_m f^{-m} \mapsto \sum_{i=0}^{\infty} \sigma(g_m) \binom{-m}{i} (\sigma(f) - f^p)^i f^{-p(m+i)}.$$

The induced (linear) action of $\sigma^a$ on MW cohomology computes $Z(Z_f, T)$. More precisely, for

$$H^n := \Omega^n / d(\Omega^{n-1}),$$

we have

$$Z(Z_f, T) = \frac{1}{(1 - T)(1 - qT) \cdots (1 - q^{n-1}T)} P_f(T)^{(-1)^n}$$

$$P_f(T) = \det(1 - q^{-1} T \sigma^a, H^n \otimes_{\mathbb{Z}_q} \mathbb{Q}_q).$$
Action of Frobenius

Define a (semilinear) endomorphism $\sigma$ of $S^\dagger$ as the absolute Frobenius lift on $R$, the substitution $[v] \mapsto [v]^p$ on monomials, and

$$g_m f^{-m} \mapsto \sum_{i=0}^{\infty} \sigma(g_m) \binom{-m}{i} (\sigma(f) - f^p)^i f^{-p(m+i)}.$$

The induced (linear) action of $\sigma^a$ on MW cohomology computes $Z(Z_f, T)$. More precisely, for

$$H^n := \Omega^n / d(\Omega^{n-1}),$$

we have

$$Z(Z_f, T) = \frac{1}{(1 - T)(1 - qT) \cdots (1 - q^{n-1}T)} P_f(T)^{(-1)^n}$$

$$P_f(T) = \det(1 - q^{-1} T \sigma^a, H^n \otimes_{\mathbb{Z}_q} \mathbb{Q}_q).$$
Griffiths-Dwork reduction

To compute the action of $\sigma^a$ on the finite-dimensional $\mathbb{Q}_q$-vector space $H^n \otimes_{\mathbb{Z}_q} \mathbb{Q}_q$, we choose a basis, apply $\sigma^a$ to each basis element, truncate the infinite sum somewhere, then reduce the result in cohomology. One way to do this is the Griffiths-Dwork reduction: for

$$\omega = d\log[e_1] \wedge \cdots \wedge d\log[e_n],$$

for $g_m \in P_m, \lambda \in L^\vee$ we have

$$\frac{g_m f}{f^{m+1}} \omega \equiv \frac{g_m}{f^m} \omega$$

$$\frac{g_m \partial_\lambda(f)}{f^{m+1}} \omega \equiv \frac{1}{m} \left( \frac{\partial_\lambda(g_m)}{f^m} \omega \right) \quad (m > 0).$$

Using a theorem of Macaulay, we lower the pole order to $n$ and then finish with explicit linear algebra. This recovers the AKR algorithm.

Unfortunately, this involves dense polynomials of degree $p^n$, and thus an unavoidable factor of $p^n$ in the runtime. But there is another way...
Griffiths-Dwork reduction

To compute the action of $\sigma^a$ on the finite-dimensional $\mathbb{Q}_q$-vector space $H^n \otimes_{\mathbb{Z}_q} \mathbb{Q}_q$, we choose a basis, apply $\sigma^a$ to each basis element, truncate the infinite sum somewhere, then reduce the result in cohomology. One way to do this is the Griffiths-Dwork reduction: for

$$\omega = d\log[e_1] \wedge \cdots \wedge d\log[e_n],$$

for $g_m \in P_m$, $\lambda \in L^\vee$ we have

$$\frac{g_m f}{f^{m+1}} \omega \equiv \frac{g_m}{f^m} \omega$$

$$\frac{g_m \partial_\lambda(f)}{f^{m+1}} \omega \equiv \frac{1}{m} \frac{\partial_\lambda(g_m)}{f^m} \omega \quad (m > 0).$$

Using a theorem of Macaulay, we lower the pole order to $n$ and then finish with explicit linear algebra. This recovers the AKR algorithm.

Unfortunately, this involves dense polynomials of degree $p n$, and thus an unavoidable factor of $p^n$ in the runtime. But there is another way...
A word on precision

Since the reduction process involves denominators, truncating $\sigma$ modulo $p^N$ does not guarantee correct computation of the matrix of action modulo $p^N$.

However, the loss of precision is bounded above by $n \log(pN)$, so the necessary working precision is not much larger than the sufficient final precision. We will hereafter ignore the distinction between the two. (It is particularly easy to analyze the situation when $p > n$.)
A word on precision

Since the reduction process involves denominators, truncating \( \sigma \) modulo \( p^N \) does not guarantee correct computation of the matrix of action modulo \( p^N \).

However, the loss of precision is bounded above by \( n \log(pN) \), so the necessary working precision is not much larger than the sufficient final precision. We will hereafter ignore the distinction between the two. (It is particularly easy to analyze the situation when \( p > n \).)
A sparse representation of Frobenius

Note that modulo $p^N$,

$$
\sigma \left( \frac{g_m}{f^m} \right) \equiv \sigma(g_m) \sum_{i=0}^{N-1} \binom{-m}{i} \left( \sigma(f) - fp \right)^i f^{-p(m+i)}
$$

$$
= \sigma(g_m) \sum_{i=0}^{N-1} \binom{-m}{i} f^{-p(m+i)} \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} \sigma(f)^j f^p(i-j)
$$

$$
= \sigma(g_m) \sum_{j=0}^{N-1} \binom{-m}{j} \sigma(f)^j f^{-p(m+j)} \sum_{i=j}^{N-1} \binom{m+i-1}{m+j-1}
$$

$$
= \sum_{j=0}^{N-1} \binom{-m}{j} \binom{m+N-1}{N-j-1} \sigma(g_m f^j) f^{-p(m+j)}.
$$

The last expression is no longer the truncation of a $p$-adically convergent series, but no matter; it involves only $p$-th power monomials!
A sparse representation of Frobenius

Note that modulo $p^N$,

$$\sigma \left( \frac{g_m}{f_m} \right) \equiv \sigma(g_m) \sum_{i=0}^{N-1} \binom{-m}{i} \left( \sigma(f) - fp \right)^i f^{-p(m+i)}$$

$$= \sigma(g_m) \sum_{i=0}^{N-1} \binom{-m}{i} f^{-p(m+i)} \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} \sigma(f)^j f^{p(i-j)}$$

$$= \sigma(g_m) \sum_{j=0}^{N-1} \binom{-m}{j} \sigma(f)^j f^{-p(m+j)} \sum_{i=j}^{N-1} \binom{m+i-1}{m+j-1}$$

$$= \sum_{j=0}^{N-1} \binom{-m}{j} \binom{m+N-1}{N-j-1} \sigma(g_m f^j) f^{-p(m+j)}.$$
Controlled reduction

By the nondegeneracy hypothesis, we can construct linear maps
\( \pi_0, \ldots, \pi_n : P_{n+1} \to P_n \) such that

\[
P_{n+1}(g_{n+1}) = \pi_0(g_{n+1})f + \sum_{i=1}^{n} \pi_i(g_{n+1}) \partial_e^*(f).
\]

Then for any \( m, j \geq 0 \) and any monomials \( \mu \in P_1, \nu \in P_m \),

\[
\frac{g_{n+1} \mu^{j+1} \nu}{f^{m+n+j+1}} \omega \equiv (m + n + j)^{-1} (R_{\mu, \nu}(g_n) + jS_{\mu}(g_n)) \frac{\mu^j \nu}{f^{m+n+j}} \omega
\]

for

\[
R_{\mu, \nu}(x) := (m + n)\pi_0(\mu x) + \sum_{h=1}^{n} (\partial_e^* + e_h^*(\nu))(\pi_h(\mu x))
\]

\[
S_{\mu}(x) := \pi_0(\mu x) + \sum_{h=1}^{n} e_h^*(\mu)\pi_h(\mu x).
\]
**Controlled reduction**

By the nondegeneracy hypothesis, we can construct linear maps \( \pi_0, \ldots, \pi_n : P_{n+1} \to P_n \) such that

\[
P_{n+1}(g_{n+1}) = \pi_0(g_{n+1})f + \sum_{i=1}^{n} \pi_i(g_{n+1}) \partial e_i^*(f).
\]

Then for any \( m, j \geq 0 \) and any monomials \( \mu \in P_1, \nu \in P_m, \)

\[
\frac{g_n \mu^{j+1} \nu}{f_{m+n+j+1}} \omega \equiv (m + n + j)^{-1}(R_{\mu, \nu}(g_n) + jS_{\mu}(g_n)) \frac{\mu^{j} \nu}{f_{m+n+j+1}} \omega
\]

for

\[
R_{\mu, \nu}(x) := (m + n)\pi_0(\mu x) + \sum_{h=1}^{n} (\partial e_h^* + e_h^* (\nu))(\pi_h(\mu x))
\]

\[
S_{\mu}(x) := \pi_0(\mu x) + \sum_{h=1}^{n} e_h^*(\mu)\pi_h(\mu x).
\]
More on controlled reduction

We thus can strip out $\mu^p$ by multiplying together $p$ matrices of size

$$\#(n\Delta \cap L) \sim n^n \text{Vol}(\Delta).$$

With a slightly more involved process, we can reduce the matrix size to $n! \text{Vol}(\Delta)$, saving a factor of $(n^n/n!) \sim e^n$.

In case $P$ is generated in degree 1, we can use controlled reduction to completely simplify the expressions occurring in the sparse Frobenius expansion.

Otherwise, the only issue is caused by monomials of the form $\sigma(g_m)$ for $m \in \{1, \ldots, n\}$. This can be resolved in various ways, e.g., by writing a small power of $g_m$ as a product of degree 1 monomials.

In any case, one must do some residual linear algebra at the end to reduce the matrix to the correct size (roughly a factor of $n$). For instance, for a quartic K3 surface, one must reduce the matrix size from 64 to 21.
A bit of complexity analysis

Unless \( \log_p q \) is large, the dominant factor is the rounds of controlled reduction. The number of such rounds is

\[
\#((n + N) \Delta \cap L) \sim (n + N)^n \text{Vol}(\Delta)
\]

Each round involves multiplying \( p \) matrices of size \( n! \text{Vol}(\Delta) \), so with straightforward matrix arithmetic we have \( O(p(n + N)^n(n!)^3 \text{Vol}(\Delta)^4) \) arithmetic operations. Note that the dependence on \( p \) is linear! (Warning: one must also factor in the \( p \)-adic precision.)

One can easily adapt for square-root dependence in \( p \) or average polynomial time dependence in \( \log p \), but we have not attempted this.
A numerical example

This example computed by Edgar Costa (NYU) using C++/NTL.

Take $n := 3$, $\Delta := \operatorname{Conv}(0, 4e_1, 4e_2, 4e_3)$. Write $x_0, x_1, x_2, x_3$ for $[0], [e_1], [e_2], [e_3]$ and put

$$f := 25163x_0^4 + 9405x_0^3x_1 + 85x_0^2x_1^2 + 30034x_0x_1^3 + 21740x_1^4$$
$$+ 14747x_0^3x_2 + 35394x_0^2x_1x_2 + 13683x_0x_1^2x_2 + 12720x_1^3x_2$$
$$+ 36331x_0^2x_2^2 + 23023x_0x_1x_2^2 + 25667x_1^2x_2^2 + 7066x_0x_2^3 + 6479x_1x_2^3$$
$$+ 8778x_2^4 + 40922x_0^3x_3 + 38119x_0^2x_1x_3 + 48775x_0x_1^2x_3 + 9720x_1^3x_3$$
$$+ 20633x_0^2x_2x_3 + 41354x_0x_1x_2x_3 + 31769x_1^2x_2x_3 + 32904x_0x_2^2x_3$$
$$+ 49443x_1x_2^2x_3 + 24957x_2^3x_3 + 37766x_0^2x_3^2 + 8622x_0x_1x_3^2 + 3377x_1^2x_3^2$$
$$+ 15688x_0x_2x_3^2 + 10170x_1x_2x_3^2 + 19668x_2^2x_3^2 + 2486x_0x_3^3 + 13807x_1x_3^3$$
$$+ 15264x_2x_3^3 + 27566x_3^4.$$

Then $Z_f$ is a nondegenerate quartic K3 surface in $\mathbb{P}^3_\mathbb{Q}$.
A numerical example (continued)

Take $p := 49999$. In 5h45m on a single-core 2.6GHz Intel Xeon (Sandy Bridge), one computes

$$P_2(Z_f, T) = 1 + a_1 T + a_2 p T^2 + \cdots + a_{10} p^9 T^{10}$$
$$- a_{10} p^{10} T^{11} - \cdots - a_2 p^{18} T^{19} - a_1 p^{19} T^{20} + p^{21}$$

with

$$(a_1, \ldots, a_{10}) = (33264, -81893, -32490, 86146, 23017, -55214, -22632, -2392, 43164, 47726).$$

This has roots in $\mathbb{C}$ as predicted by the Weil conjectures (see Sage notebook).