Zeta functions of nondegenerate hypersurfaces in toric varieties via controlled reduction in $p$-adic cohomology

Edgar Costa, David Harvey and Kiran S. Kedlaya
Zeta functions of nondegenerate hypersurfaces in toric varieties via controlled reduction in $p$-adic cohomology

Edgar Costa, David Harvey and Kiran S. Kedlaya

We give an interim report on some improvements and generalizations of the Abbott–Kedlaya–Roe method to compute the zeta function of a nondegenerate ample hypersurface in a projectively normal toric variety over $\mathbb{F}_p$ in linear time in $p$. These are illustrated with a number of examples including K3 surfaces, Calabi–Yau threefolds, and a cubic fourfold. The latter example is a nonspecial cubic fourfold appearing in the Ranestad–Voisin coplanar divisor on moduli space; this verifies that the coplanar divisor is not a Noether–Lefschetz divisor in the sense of Hassett.

1. Introduction

We consider the problem of computing the zeta function $Z(\mathcal{X}, t)$ of an explicitly specified variety $\mathcal{X}$ over a finite field $\mathbb{F}_q$ of characteristic $p$. For curves and abelian varieties, Schoof’s method and variants [Sch85; Pil90; GS04; GKS11; GS12] can compute $Z(\mathcal{X}, t)$ in time and space polynomial in $\log q$ and exponential in the genus/dimension; these have only been implemented for genus/dimension at most 2. Such methods may be characterized as $\ell$-adic, as they access the $\ell$-adic cohomology (for $\ell \neq p$ prime) of the variety via torsion points; there also exist $p$-adic methods which compute approximations of the Frobenius action on $p$-adic cohomology (Monsky–Washnitzer cohomology), and which have proven to be more viable in practice for large genus. Early examples include Kedlaya’s algorithm [Ked01] for hyperelliptic curves, in which the time/space dependence is polynomial in the genus and quasilinear in $p$, and Harvey’s algorithm [Har07], which improves the dependence on $p$ to $p^{1/2+\epsilon}$. These methods have been subsequently generalized [GG01; DV06a; DV06b; Har12], notably by Tuitman’s algorithm [Tui16; Tui17], which applies to (almost) all curves while keeping the quasilinear dependence on $p$. In another
direction, Harvey [Har14] has shown that when computing the zeta functions of reductions of a fixed hyperelliptic curve over a number field, $p$-adic methods can achieve average polynomial time in $\log p$ and the genus; this has been implemented in small genus [HS14; HS16].

One advantage of $p$-adic methods over $\ell$-adic ones is that they scale much better to higher-dimensional varieties. For example, there are several $p$-adic constructions that apply to arbitrary varieties with reasonable asymptotic complexity [LW08; Har15], although we are not aware of any practical implementations. Various algorithms, and some implementations, have been given using Lauder’s deformation method of computing the Frobenius action on the Gauss–Manin connection of a pencil [Lau04a; Lau04b; Ger07; Hub08; PT15; Tui19].

In this paper, we build on an algorithm of Abbott, Kedlaya and Roe [AKR10] which adapts the original approach of [Ked01] to smooth projective hypersurfaces. Here, we add two key improvements:

- We use controlled reduction in de Rham cohomology, as described in some lectures of Harvey [Har10a; Har10b; Har10c], to preserve sparsity of certain polynomials, thus reducing the time (respectively, space) dependence on $p$ from polynomial to quasilinear (respectively, $O(\log p)$). The resulting controlled AKR method was implemented, with further improvements, in Costa’s Ph.D. thesis [Cos15], with examples of generic surfaces and threefolds over $\mathbb{F}_p$ for $p \sim 10^6$ [Cos15, §1.6]; by contrast, the largest $p$ used in [AKR10] is 29. Costa and Harvey are currently preparing a paper on this method; meanwhile, Costa’s GPL-licensed code is available on GitHub [Cos] and is slated to be integrated into SageMath [Sag].

- We also generalize to toric hypersurfaces, subject to a standard genericity condition called nondegeneracy. This greatly increases the applicability of the method while preserving much of its efficiency. Some previous attempts have been made to compute zeta functions in this setting, such as work of Castryck, Denef and Vercauteren [CDV06] for curves and Sperber and Voight [SV13] in general; it is the combination with controlled reduction that makes our approach the most practical to date.

It may be possible to improve the dependence on $p$ to square-root (as in [Har07]) or average polynomial time (as in [Har14]), but we do not attempt to do so here.

For reasons of space, we give only a summary of the algorithm, with further details to appear elsewhere. In lieu of these details, we present a number of worked examples in dimensions 2–4 that demonstrate the practicality of this algorithm in a wide range of cases. The results are based on an implementation in C++, using NTL [Sho] for the underlying arithmetic operations. Our examples in dimensions 2 and 3 were computed on one core of a desktop machine with an Intel Core i5-4590 3.30GHz processor; our sole example in dimension 4 was computed on one core of a server with an AMD Opteron 6378 1.6GHz processor. (We have not yet optimized our vector-matrix multiplications in any way; as a consequence, we observe a serious performance hit whenever the working moduli exceeds $2^{62}$.)

In dimensions 2 and 3, our examples are Calabi–Yau varieties, i.e., smooth, proper, simply connected varieties with trivial canonical bundle. In dimension 1, these are simply elliptic curves. In dimension 2, they are K3 surfaces, whose zeta functions are of computational interest for various reasons. For instance,
these zeta functions can (potentially) be used to establish the infinitude of rational curves on a K3 surface (see the introduction to [CT14] for discussion); there has also been recent work on analogues of the Honda–Tate theorem, establishing conditions under which particular zeta functions are realized by K3 surfaces [Tae16; Ito16].

As for Calabi–Yau threefolds, much of the interest in their zeta functions can be traced back to mirror symmetry in mathematical physics. An early example is the work of Candelas, de la Ossa and Rodriguez Villegas [CdlORV03] on the Dwork pencil; a more recent example is [DKS+16], in which (using $p$-adic cohomology) certain mirror families of Calabi–Yau threefolds are shown to have related zeta functions.

Our four-dimensional example is a cubic projective fourfold. Such varieties occupy a boundary region between rational and irrational varieties; it is expected that a rational cubic fourfold is special in the sense of having a primitive cycle class in codimension 2. The geometry of special cubic fourfolds is in turn closely linked to that of K3 surfaces; in many cases, the Hodge structure of a K3 surface occurs (up to a twist) inside the Hodge structure of a special cubic fourfold, and (modulo standard conjectures) this implies a similar relationship between zeta functions. See [Has16] for further discussion.

The specific example we consider is related to the geometry of the moduli space of cubic fourfolds over $\mathbb{C}$. On this space, there exist various divisors consisting entirely of special cubic fourfolds; Hassett calls these Noether–Lefschetz divisors (by analogy with the case of surfaces). Recently, Ranestad and Voisin [RV17] exhibited four divisors which they believed not to be Noether–Lefschetz, but only checked this in one case. Addington and Auel [AA17] checked two more cases by finding in these divisors some cubic fourfolds over $\mathbb{Q}$ with good reduction at 2 such that the zeta functions over $\mathbb{F}_2$ show no primitive Tate classes in codimension 2. By replacing the brute-force point counts of Addington and Auel with $p$-adic methods, we are able to work modulo a larger prime to find an example showing that the fourth Ranestad–Voisin divisor is not Noether–Lefschetz.

To sum up, the overall goal of this project is to vastly enlarge the collection of varieties for which computing the zeta function is practical. It is our hope that doing so will lead to a rash of new insights, conjectures, and theorems of interest to a broad range of number theorists and algebraic geometers.

2. Toric hypersurfaces

We begin by reviewing the construction of a projective toric variety from a lattice polytope. For more details we recommend [CLS11].

Let $n \geq 1$ be an integer. For any commutative ring $R$, let $R[x^\pm]$ denote the Laurent polynomial ring in $n$ variables $x_1, \ldots, x_n$ with coefficients in $R$. For $\alpha := (\alpha_i) \in \mathbb{Z}^n$, we write $x^\alpha$ for the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. We denote the $R$-torus by $T^n_R := \text{Spec}(R[x^\pm])$.

Let $\Delta \subset \mathbb{R}^n$ be the convex hull of a finite subset of $\mathbb{Z}^n$ that is not contained in any hyperplane, so that dim $\Delta = n$. For $r \in \mathbb{R}$, let $r\Delta$ be the $r$-fold dilation of $\Delta$. For an integer $d \geq 0$, let

$$P_d := \langle x^\alpha : \alpha \in d\Delta \cap \mathbb{Z}^n \rangle_R \quad \text{and} \quad P_{d}^{\text{Int}} := \langle x^\alpha : \alpha \in \text{Int}(d\Delta) \cap \mathbb{Z}^n \rangle_R$$
be the free $R$-modules on the sets of monomials with exponents in $d \Delta \cap \mathbb{Z}^n$ and $\text{Int}(d \Delta) \cap \mathbb{Z}^n$ respectively. Define the $R$-graded algebras
\[ P_\Delta := \bigoplus_{d=0}^{+\infty} P_d \quad \text{and} \quad P_\Delta^{\text{Int}} := \bigoplus_{d=0}^{+\infty} P_d^{\text{Int}} \]
with the usual multiplication in $R[x^\pm]$. We define the polarized toric variety associated to $\Delta$ as the pair $(\mathbb{P}_\Delta, \mathcal{O}_\Delta)$, where $\mathbb{P}_\Delta := \text{Proj} P_\Delta$ and $\mathcal{O}_\Delta$ is the ample line bundle on $\mathbb{P}_\Delta$ associated to the graded $P_\Delta$-module $P_\Delta(1)$. Note that $P_\Delta$ and $P_\Delta^{\text{Int}}$ admit $n$ commuting degree-preserving differential operators $\partial_i := x_i(\partial/\partial x_i)$ for $i = 1, \ldots, n$.

In order to suppress some expository and algorithmic complexity, we make the simplifying assumption that $\Delta$ is a normal polytope; that is, the map
\[ (\Delta \cap \mathbb{Z}^n)^d \to d \Delta \cap \mathbb{Z}^n, \quad (x_1, \ldots, x_d) \mapsto x_1 + \cdots + x_d, \]
is surjective for $d \geq 1$. This corresponds to the pair $(\mathbb{P}_\Delta, \mathcal{O}_\Delta)$ being projectively normal; this will be the case in our examples. As a consequence, we have that $\mathcal{O}_\Delta$ is indeed very ample.

**Example 2.1.** Let $\Delta$ be the regular $n$-simplex, the convex hull of $0, e_1, \ldots, e_n$. We may then identify $P_d$ with the set of homogeneous polynomials of degree $d$ in $x_0, \ldots, x_n$, by identifying $x^\alpha \in P_{\Delta,d}$ with the monomial $x_0^{d-\alpha_1} \cdots x_n^{\alpha_n}$; then $(\mathbb{P}_\Delta, \mathcal{O}_\Delta)$ is isomorphic to $(\mathbb{P}_R^n, \mathcal{O}(1))$.

We obtain the weighted projective space $\mathbb{P}(w_0, \ldots, w_n)$ by taking
\[ \Delta := \left\{ (x_0, \ldots, x_n) \in \mathbb{R}^{n+1} : \sum_{i=0}^n w_i x_i = w_0 \cdots w_n \right\}; \]
see [Dol82, 1.2.5].

We obtain $\mathbb{P}_R^k \times_R \mathbb{P}_R^r$ by taking $\Delta$ to be the Cartesian product of the regular $k$-simplex by the regular $r$-simplex [CLS11, §2.4].

We now turn our attention to toric hypersurfaces over $R = \mathbb{F}_q$, the finite field with $q = p^a$ elements and characteristic $p$. Let $\mathcal{Y}$ be the hypersurface in $\mathbb{T}_q^n$ defined by a Laurent polynomial $\tilde{f} \in \mathbb{F}_q[x^\pm]$, $\mathcal{Y} := V(\tilde{f}) \subset \mathbb{T}_q^n$. Let
\[ \text{supp} \, \tilde{f} = \{ \alpha \in \mathbb{Z}^n : \overline{c}_\alpha \neq 0 \} \]
be the support of $\tilde{f}$ in $\mathbb{R}^n$; the convex hull of $\text{supp} \, \tilde{f}$ is the Newton polytope of $\tilde{f}$, which we denote by $\Delta$. We will work under the hypothesis that $\tilde{f}$ is $(\Delta)$-nondegenerate:¹ for all faces $\tau \subseteq \Delta$ (including $\Delta$ itself), the system of equations
\[ \tilde{f}|_\tau = \partial_1 \tilde{f}|_\tau = \cdots = \partial_n \tilde{f}|_\tau = 0 \]
has no solution in $\mathbb{F}_q^n$, where $\mathbb{F}_q$ denotes an algebraic closure of $\mathbb{F}_q$. Furthermore, nondegeneracy implies quasismoothness; see [BC94, Definition 3.1 and Proposition 4.15]. For fixed normal $\Delta$ over an

¹This condition was introduced by Dwork [Dwo62] without a name; the term nondegenerate first appears in [Kou76; Var76]. Synonyms include $\Delta$-regular [Bat93, § 4] and schön [Tev07].
infinite field, this condition holds for generic $\bar{f}$. Others have given point-counting algorithms under this assumption [CDV06; SV13].

Let $\mathcal{X} := \text{Proj } P_\Delta/(\bar{f})$ denote the closure of $\mathcal{Y}$ in $\mathbb{P}_\Delta$ (placing $\bar{f}$ in degree 1) and set $\mathcal{U} := \mathbb{T}^n \setminus \mathcal{Y}$. Let $H^i_{\text{rig}}$ denote the $i$-th rigid cohomology group in the sense of Berthelot [Ber97]. The Lefschetz hyperplane theorem, combined with Poincaré duality, shows that the map

$$H^i_{\text{rig}}(\mathbb{P}_\Delta) \rightarrow H^i_{\text{rig}}(\mathcal{X})$$

induced by the inclusion $\mathcal{X} \hookrightarrow \mathbb{P}_\Delta$ is an isomorphism for $i \neq n - 1$ [BC94, Proposition 10.8]. This implies that the “interesting” part of the cohomology of $\mathcal{X}$ occurs in dimension $n - 1$ and consists of those classes that do not come from $\mathbb{P}_1$. Denote by $\text{PH}^{n-1}_{\text{rig}}(\mathcal{X})$ the primitive cohomology group of $\mathcal{X}$, defined by the (Frobenius-equivariant) exact sequence

$$0 \rightarrow H^{n-1}_{\text{rig}}(\mathbb{P}_\Delta) \rightarrow H^{n-1}_{\text{rig}}(\mathcal{X}) \rightarrow \text{PH}^{n-1}_{\text{rig}}(\mathcal{X}) \rightarrow 0.$$

With this notation, we may write

$$Z(\mathcal{X}, t) = Z(\mathbb{P}_\Delta, t)Q(t)^{(-1)^n},$$

where

$$Q(t) := \det(1 - t \text{Frob}_q \mid \text{PH}^{n-1}_{\text{rig}}(\mathcal{X})).$$

Thus given $\bar{f}$, we would like to compute $Q(t)$.

The cohomology group $\text{PH}^{n-1}_{\text{rig}}(\mathcal{X})$ is closely related to $H^n_{\text{rig}}(\mathbb{P}_\Delta \setminus \mathcal{X})$. For example, if $\mathbb{P}_\Delta$ is a (weighted) projective space, as in [AKR10; Cos15], the two cohomology groups are isomorphic; see [BC94, Proposition 10.11].

### 3. de Rham cohomology of toric hypersurfaces

In preparation for our use of $p$-adic cohomology to compute $Q(t)$, we give an explicit description of the algebraic de Rham cohomology of a nondegenerate toric hypersurface in characteristic zero. We take $R$ to be the ring $\mathbb{Z}_q$, that is, the ring of integers of $\mathbb{Q}_q$, which is the unramified extension of $\mathbb{Q}_p$ with residue field $\mathbb{F}_q$.

Let $f \in \mathbb{Z}_q[x^\pm]$ be a lift of $\bar{f}$ to characteristic zero with the same support as $f$ (it will also be nondegenerate). Consider $Y := V(f) \subset \mathbb{T} := \mathbb{T}_{\mathbb{Q}_q}$ and $X$, the closure of $Y$ in $\mathbb{P}_\Delta$. Write $U := \mathbb{T} \setminus Y$, and $V := \mathbb{P}_\Delta \setminus X \simeq \text{Spec}(A)$, where $A$ is the coordinate ring of $V$; explicitly,

$$A \simeq \bigcup_{d=0}^{+\infty} f^{-d} P_d.$$

Let $I_f$ be the ideal in $P_\Delta$ generated by $f, \partial_1 f, \ldots, \partial_n f$. We call $I_f$ the toric Jacobian ideal and the quotient ring $J_f := P_\Delta/I_f$ the toric Jacobian ring. Since $f$ is nondegenerate, the ideal $I_f$ is irrelevant in $P_\Delta$ and $\text{rank}_{\mathbb{Z}_q} J_f = n! \text{Vol}(\Delta)$; furthermore, $(J_f)_d = 0$ for $d > n$ [Bat93, §4]. If $\mathcal{O}_\Delta$ is not very ample, then $I_f$ might not be generated in degree 1 and we might have $(J_f)_d = 0$ only for $d \gg n$. 

Let $\Omega^\bullet$ denote the logarithmic de Rham complex of $V$ with poles along $\mathbb{P}^1 \setminus \mathbb{T}$. Let $H^\bullet$ be the cohomology groups of $\Omega^\bullet$; these are naturally isomorphic to $H^\bullet_{\text{dR}}(V \cap \mathbb{T} = \mathbb{T} \setminus Y = U)$ and $H^\bullet_{\text{rig}}(\mathbb{T}_F \setminus Y = \mathcal{U})$ [Kat89].

We now provide an explicit description of the group $H^n$, as in [Bat93, §§6 and 7], in which we will compute $Q(t)$. Set

$$
\omega := \frac{dx_1}{x_1} \wedge \cdots \wedge \frac{dx_n}{x_n} \in \Omega^n,
$$

and define the ascending filtration in $\Omega^n$ by

$$
\text{Fil}^d \Omega^n := \{gf^{-d}\omega : g \in P_d\}.
$$

The associated graded ring

$$
\Omega^n := \bigoplus_{d=0}^{\infty} \text{Gr}^d \Omega^n, \quad \text{Gr}^d \Omega^n := \text{Fil}^d \Omega^n / \text{Fil}^{d-1} \Omega^n,
$$

is then isomorphic to $P_{\Delta}/(f)$ (again placing $f$ in degree 1).

Equip $H^n$ with the filtration induced from $\Omega^n$, and view $H^n$ as the quotient of $\Omega^n$ by the $\mathbb{Q}_d$-submodule generated by the relations

$$
\frac{g}{f^d} \omega = \frac{gf}{f^{d+1}} \omega \quad \text{and} \quad \frac{\partial_i(g)}{f^d} \omega = \frac{d \partial_i(f)}{f^{d+1}} \omega
$$

for each $i = 1, \ldots, n$, each nonnegative integer $d$, and each $g \in P_d$. From these relations, we see that

$$
\text{Gr}^1 H^n \simeq P_1/(f) \quad \text{and} \quad \text{Gr}^d H^n \simeq (J_f)_d \quad (d > 1).
$$

This gives a way to compute explicitly in $H^n$: for any $h \in (J_f)_{d+1}$ with $d > n$, we can find a relation of the form

$$
\frac{d}{f^{d+1}} \omega = \frac{d g_0 f + \sum_{i=1}^{n} g_i \partial_i f}{f^{d+1}} \omega = \frac{d \partial_0 g + \sum_{i=1}^{n} \partial_i g_i}{f^d} \omega
$$

because $P_d \subset (J_f)_d$, so in $H^n$ we can reduce the pole order of any form to at most $n$. This process was introduced for smooth projective hypersurfaces in [Gri69a; Gri69b] and attributed to Dwork; it is commonly known as Griffiths–Dwork reduction.

With the above representation of $H^n$, we may also identify $PH^1_{\text{dR}}(X)$ with $(P^\text{Int}_{\Delta} + I_f)/I_f \subset H^n$, where the filtration by pole order is the Hodge filtration; see [Bat93; BC94, §§9 and 11].

We now introduce a variation of Griffiths–Dwork reduction, called controlled reduction. This will be crucial for our application to $p$-adic cohomology, as careless application of Griffiths–Dwork reduction to a sparse form will easily lead to a dense form. For $d = 1, \ldots, n+1$, choose a $\mathbb{Z}_d$-linear splitting $P_d \approx (J_f^\prime)_d \oplus C_d$, where $(J_f^\prime)_d$ is a lift of $(J_f)_d$ into $P_d$. Let $\rho_d: P_d \rightarrow (J_f^\prime)_d$ and $\pi_{d,0}, \ldots, \pi_{d,n}: P_d \rightarrow P_{d-1}$ be $\mathbb{Z}_d$-linear maps such that

$$
g = \rho_d(g) + \pi_{d,0}(g) \cdot f + \sum_{i=1}^{n} \pi_{d,i}(g) \cdot \partial_i f, \quad g \in P_d.
$$

These maps may be constructed one monomial at a time.
Proposition 3.1 (controlled reduction). Let \( x^\nu \in P_1 \) and \( x^\mu \in P_d \) be two monomials and define the \( \mathbb{Z}_q \)-linear maps
\[
R_{\mu,\nu}(g) := (d + n)\pi_{n+1,0}(x^\nu g) + \sum_{i=1}^n (\partial_i + \mu_i)(\pi_{n+1,i}(x^\nu g)),
\]
\[
S_{\nu}(g) := \pi_{n+1,0}(x^\nu g) + \sum_{i=1}^n v_i\pi_{n+1,i}(x^\nu g).
\]
Then for any \( g \in P_n \) and any nonnegative integer \( j \), in \( H^n \) we have
\[
g \frac{x^{(j+1)\nu+\mu}}{f^{d+n+j+1}} \omega \equiv (d + n + j)^{-1}(R_{\mu,\nu}(g) + jS_{\nu}(g)) \frac{x^{j\nu+\mu}}{f^{d+n+j}} \omega.
\]

Proof. This is straightforward from (3-1) and (3-2). \( \square \)

Note that Proposition 3.1 enables us to reduce the pole order of a differential form from \( d + n + j + 1 \) to \( d + n + j \) without increasing its total number of monomials; we can thus reduce the pole order of a sparse form without making it dense.

Corollary 3.2. With notation as in Proposition 3.1, let \( k \) be a positive integer. Then for any \( g \in P_n \),
\[
g \frac{x^{\mu+k\nu}}{f^{d+n+k}} \omega \equiv \frac{\prod_{j=0}^{k-1}(R_{\mu,\nu} + jS_{\nu})(g)}{\prod_{j=0}^{k-1}(d + n + j)} \frac{x^{\mu}}{f^{d+n}} \omega,
\]
forming the composition product from left to right.

Using Proposition 3.1 amounts to performing linear algebra on matrices of size \( #(n\Delta \cap \mathbb{Z}^n) \sim n^n \text{Vol}(\Delta) \). One can reduce this by a factor of \( n^n/n! \sim e^n \) at the expense of making the expression for the reduction matrix more convoluted; compare [Cos15, Remark 1.17 and Proposition 1.18].

4. Monsky–Washnitzer cohomology

We now indicate how Monsky–Washnitzer cohomology, as introduced in [MW68; Mon68; Mon71], provides a crucial link between algebraic de Rham cohomology and \( p \)-adic rigid cohomology, by transferring to the former the canonical Frobenius action on the latter; see also [vdP86]. To simplify, we assume \( p > \max\{n, 2\} \).

Let \( A^\dagger \) denote the weak \( p \)-adic completion of \( A \), the ring consisting of formal sums \( \sum_{d=0}^{+\infty} g_d f^{-d} \) such that for some \( a, b > 0 \), \( g_d \in P_{d}^{\max\{a, |d-b|\}} \) for all \( d \geq 0 \). We define the associated logarithmic de Rham complex \( \Omega^{\dagger,i} := \Omega^i \otimes_A A^\dagger \); denote the cohomology groups of this complex by \( H^{\dagger,i} \). We may then obtain \( p \)-adic Monsky–Washnitzer cohomology groups \( H^{\dagger,i} \otimes_{\mathbb{Z}_q} \mathbb{Q}_q \). The map \( \Omega^i \otimes_{\mathbb{Z}_q} \mathbb{Q}_q \to \Omega^{\dagger,i} \otimes_{\mathbb{Z}_q} \mathbb{Q}_q \) is a quasi-isomorphism [Mon70; vdP86; Kat89]; that is, the induced maps \( H^i \otimes_{\mathbb{Z}_q} \mathbb{Q}_q \to H^{\dagger,i} \otimes_{\mathbb{Z}_q} \mathbb{Q}_q \) are isomorphisms. We can thus identify the algebraic de Rham cohomology of \( U \) with the Monsky–Washnitzer cohomology of \( U \).

On the other hand, we also have \( H^{\dagger,i} \otimes_{\mathbb{Z}_q} \mathbb{Q}_q \simeq H_{\text{rig}}^\bullet (U) \) and the latter object is functorial with respect to geometry in characteristic \( p \) [Ber97]. In this way, \( H^{\dagger,i} \) receives an action of the Frobenius automorphism,
which we can make explicit by constructing a lift $\sigma$ of the $p$-th power Frobenius on $\mathbb{F}_q$ to $A^\dag$. To do so, we take the Witt vector Frobenius on $\mathbb{Z}_q$ and set $\sigma(\mu) = \mu^p$ for any monomial $\mu \in P_\Delta$. We then extend $\sigma$ to $A^\dag$ by the formula

$$\sigma \left( \frac{g f d}{h} \right) := \sigma(g) \sigma(f)^{-d} = \sigma(g) \sum_{i \geq 0} \left( \frac{-d}{i} \right) \frac{(\sigma(f) - f^p)^i}{f^{p(d+i)}}.$$  \hspace{1cm} (4-1)

The above series converges (because $p$ divides $\sigma(f) - f^p$) and the definitions ensure that $\sigma$ is a semi-linear (with respect to the Witt vector Frobenius) endomorphism of $A^\dag$. We finally extend $\sigma$ to $\Omega^{\cdot \cdot}$ by $\sigma(g d h) := \sigma(g) d(\sigma(h))$.

5. Sketch of the algorithm

We now indicate briefly how to use controlled reduction to compute the Frobenius action on the cohomology of nondegenerate toric hypersurfaces. We start as in [Har07, Proposition 4.1], by rewriting the Frobenius action in a sparser form.

**Lemma 5.1.** For any positive integers $d$, $N$ and $g \in P_d$, in $A^\dag$ we have

$$\sigma \left( \frac{g f d}{h} \right) \equiv \sum_{j=0}^{N-1} \left( \frac{-d}{j} \right) \binom{d+N-1}{d+j} \sigma(g f^j) f^{-p(d+j)} \pmod{p^N}.$$  \hspace{1cm} (4-1)

**Proof.** This follows from (4-1) by truncating the sum and then rewriting it formally; see [Cos15, Lemma 1.10]. \hfill $\Box$

In order to compute a $p$-adic approximation of the Frobenius action on $PH^{n-1}(X)$, we must first fix a basis of the latter; we do this by constructing a monomial basis for $PH^{n-1}_{\text{dR}}(X)$ via explicit linear algebra. We then apply Frobenius to each basis element in the sparse truncated form given by Lemma 5.1, recursively reduce the pole order using Corollary 3.2 (using $k = p$ as much as possible), and project to the chosen monomial basis. The dominant step is controlled reduction, which amounts to $O(pn^N \text{Vol}(\Delta))$ matrix multiplications of size $n! \text{Vol}(\Delta)$ per basis element.

We will not address precision estimates in this report, except to note that the machinery of [AKR10, §3.4] applies. In general, if we want $N$ digits of $p$-adic accuracy, we must apply Lemma 5.1 with $N$ replaced by $N' = N + O(n + \log N)$ and work modulo $p^{O(N')}$. Hence, with respect to $p$ alone, we expect our algorithm to run in quasilinear time in $p$ and use $O(\log p)$ space.

6. K3 surfaces

We now turn our attention to examples, starting with K3 surfaces. For $X$ a K3 surface, $\dim H^2(X) = 22$ and the Hodge numbers are $(1, 20, 1)$. A common example of a K3 surface is a smooth quartic surface in $\mathbb{P}^3$; but they also occur in other ways, such as hypersurfaces in weighted projective spaces. Using a criterion of Miles Reid [Rei80], Yonemura [Yon90] found the complete list of (polarized) weighted projective spaces in which a generic hypersurface is a K3 surface; there are 95 of these. For toric varieties, the corresponding classification is that of reflexive 3-dimensional polytopes, of which there are 4,319 in all [KS98].
In the following examples, we worked modulo \( p^4 \) in order to obtain \( Q(t) \) with two \( p \)-adic significant digits. As a result, we observe a performance hit for \( p > 2^{16} \).

**Example 6.1.** Consider the projective quartic surface \( \mathcal{X} \subset \mathbb{P}_\mathbb{F}_p^3 \) defined by

\[
x^4 + y^4 + z^4 + w^4 + \lambda xyzw = 0;
\]

it is a member of the Dwork pencil. For \( p = 2^{20} - 3 \) and \( \lambda = 1 \), using the controlled AKR algorithm in 22h7m we compute that

\[
Z(\mathcal{X}, t)^{-1} = (1 - t)(1 - pt)^{16}(1 + pt)^3(1 - p^2 t)Q(t),
\]

where the “interesting” factor is

\[
Q(t) = (1 + pt)(1 - 1688538t + p^2 t^2).
\]

For this family, the remaining factors, apart from \( Q(t) \), could have also been deduced by a \( p \)-adic formula of de la Ossa and Kadir [Kad04, Chapter 6]. In this context, the Hodge numbers of \( PH^2(\mathcal{X}) \) are \((1, 19, 1)\).

A similar runtime would be expected if we used our current implementation to compute \( Z(\mathcal{X}, t) \) with \( \Delta \) being the 3-simplex (tetrahedron), as indicated by the outer polytope at right. Instead, we observe that the monomials defining \( \mathcal{X} \) generate a sublattice of index 4\(^2\) in \( \mathbb{Z}^3 \); hence, we can instead run our algorithm with a polytope of significantly smaller volume (32/3 \( \approx \) 10.66 versus 2/3 \( \approx \) 0.66), as indicated by the inner polytope at right. This leads to a dramatic speedup: with our current implementation, we computed \( Q(t) \) in 1m33s. We present the running times for other \( p \) in Table 1; memory usage was about 16MB.

In the new framework, \( \mathcal{X} \) is given by the closure (in \( \mathbb{P}_\Delta \)) of the affine surface defined by the Laurent polynomial

\[
x^4 y^{-1} z^{-1} + \lambda x + y + z + 1,
\]

A similar runtime would be expected if we used our current implementation to compute \( Z(\mathcal{X}, t) \) with \( \Delta \) being the 3-simplex (tetrahedron), as indicated by the outer polytope at right. Instead, we observe that the monomials defining \( \mathcal{X} \) generate a sublattice of index 4\(^2\) in \( \mathbb{Z}^3 \); hence, we can instead run our algorithm with a polytope of significantly smaller volume (32/3 \( \approx \) 10.66 versus 2/3 \( \approx \) 0.66), as indicated by the inner polytope at right. This leads to a dramatic speedup: with our current implementation, we computed \( Q(t) \) in 1m33s. We present the running times for other \( p \) in Table 1; memory usage was about 16MB.

In the new framework, \( \mathcal{X} \) is given by the closure (in \( \mathbb{P}_\Delta \)) of the affine surface defined by the Laurent polynomial

\[
x^4 y^{-1} z^{-1} + \lambda x + y + z + 1,
\]

A similar runtime would be expected if we used our current implementation to compute \( Z(\mathcal{X}, t) \) with \( \Delta \) being the 3-simplex (tetrahedron), as indicated by the outer polytope at right. Instead, we observe that the monomials defining \( \mathcal{X} \) generate a sublattice of index 4\(^2\) in \( \mathbb{Z}^3 \); hence, we can instead run our algorithm with a polytope of significantly smaller volume (32/3 \( \approx \) 10.66 versus 2/3 \( \approx \) 0.66), as indicated by the inner polytope at right. This leads to a dramatic speedup: with our current implementation, we computed \( Q(t) \) in 1m33s. We present the running times for other \( p \) in Table 1; memory usage was about 16MB.

<table>
<thead>
<tr>
<th>( p )</th>
<th>CHK time</th>
<th>PT time</th>
<th>( p )</th>
<th>CHK time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 2^8 - 5 )</td>
<td>0.03s</td>
<td>1.65s</td>
<td>( 2^{17} - 1 )</td>
<td>11.9s</td>
</tr>
<tr>
<td>( 2^9 - 3 )</td>
<td>0.04s</td>
<td>3.64s</td>
<td>( 2^{18} - 5 )</td>
<td>23.4s</td>
</tr>
<tr>
<td>( 2^{10} - 3 )</td>
<td>0.04s</td>
<td>7.39s</td>
<td>( 2^{19} - 1 )</td>
<td>46.9s</td>
</tr>
<tr>
<td>( 2^{11} - 9 )</td>
<td>0.06s</td>
<td>14.65s</td>
<td>( 2^{20} - 3 )</td>
<td>1m 33s</td>
</tr>
<tr>
<td>( 2^{12} - 3 )</td>
<td>0.08s</td>
<td>34.80s</td>
<td>( 2^{21} - 9 )</td>
<td>3m 6s</td>
</tr>
<tr>
<td>( 2^{13} - 1 )</td>
<td>0.13s</td>
<td>34.80s</td>
<td>( 2^{22} - 3 )</td>
<td>6m 15s</td>
</tr>
<tr>
<td>( 2^{14} - 3 )</td>
<td>0.22s</td>
<td>2m 33s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{15} - 19 )</td>
<td>0.41s</td>
<td>6m 43s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{16} - 15 )</td>
<td>5.72s</td>
<td>14m 14s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** The second and fifth columns use our current implementation to compute \( Q(t) \). The third column uses the Pancratz–Tuitman implementation [PT15] to compute \( Z(\mathcal{X}, t) \).
and the Hodge numbers of $PH^2(\mathcal{X})$ are $(1, 1, 1)$, which explains why $\deg Q(t) = 3$.

Since the Dwork pencil is a “small” deformation of the Fermat quartic, we may also use the Pancratz–Tuitman implementation of the deformation method [PT15] to compute $Z(\mathcal{X}, t)$. We did this and verified that our results agree; we compare running times in Table 1. To interpret these fairly, note that Pancratz and Tuitman work in $\mathbb{P}^3$ and so compute the whole numerator of $Z(\mathcal{X}, t)$ rather than just $Q(t)$. (Note that the algorithm of [Tui19] has a square-root dependence on $p$, as in [Har07].)

**Example 6.2.** Consider the projective quartic surface $\mathcal{X} \subset \mathbb{P}^3_{\mathbb{F}_p}$ defined by

$$x^3y + y^4 + z^4 + w^4 - 12xyzw,$$

which contains a hypergeometric motive; see [DKS16, §5]. For $p = 2^{15} - 19$, using the controlled AKR algorithm, in $27m 12s$ we compute that

$$Z(\mathcal{X}, t)^{-1} = (1 - t)(1 - pt)^2(1 + pt)^2(1 - pt + p^2t^2)^2(1 - p^2t^2 + p^4t^4)^2(1 - p^2t)Q(t),$$

where the “interesting” factor is (up to rescaling)

$$p Q(t/p) = p + 20508t^1 - 18468t^2 - 26378t^3 - 18468t^4 + 20508t^5 + pt^6.$$

As in the previous example, the Newton polytope has volume 8, but the defining monomials generate a sublattice of index 4 in $\mathbb{Z}^3$; we may thus work instead with a polytope of volume 2 (depicted at left) and observe a significant speedup. In this setting, the Hodge numbers of $PH^2(\mathcal{X})$ are $(1, 4, 1)$. With our current implementation we computed $Q(t)$ in $4s$. We present the running times for other $p$ in Table 2, where the memory footprint was about 52MB.

Alternatively, one could try to use Magma [BCP97] to confirm $Q(t)$. Unfortunately, Magma is only able to confirm the linear coefficient:

```plaintext
> C2F2 := HypergeometricData([6,12], [1,1,1,2,3]);
> EulerFactor(C2F2, 2^10 * 3^6, 2^15 - 19: Degree:=1);
1 + 20508*$.1 + O($.1^2)
```

<table>
<thead>
<tr>
<th>$p$</th>
<th>$Q(t)$ time</th>
<th>$p$</th>
<th>$Q(t)$ time</th>
<th>$p$</th>
<th>$Q(t)$ time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^8 - 5$</td>
<td>0.20s</td>
<td>$2^{13} - 1$</td>
<td>1.12s</td>
<td>$2^{18} - 5$</td>
<td>4m 54s</td>
</tr>
<tr>
<td>$2^9 - 3$</td>
<td>0.23s</td>
<td>$2^{14} - 3$</td>
<td>2.08s</td>
<td>$2^{19} - 1$</td>
<td>9m 46s</td>
</tr>
<tr>
<td>$2^{10} - 3$</td>
<td>0.29s</td>
<td>$2^{15} - 19$</td>
<td>4.00s</td>
<td>$2^{20} - 3$</td>
<td>19m 32s</td>
</tr>
<tr>
<td>$2^{11} - 9$</td>
<td>0.41s</td>
<td>$2^{16} - 15$</td>
<td>1m 11s</td>
<td>$2^{21} - 9$</td>
<td>38m 58s</td>
</tr>
<tr>
<td>$2^{12} - 3$</td>
<td>0.64s</td>
<td>$2^{17} - 1$</td>
<td>2m 30s</td>
<td>$2^{22} - 3$</td>
<td>1h 18m</td>
</tr>
</tbody>
</table>

**Table 2.** Running times for Example 6.2.
Example 6.3. Consider the closure $\mathcal{X}$ in $\mathbb{P}_3$ (which in this case is not a weighted projective space) of the affine surface defined by the Laurent polynomial

$$3x + y + z + x^{-2}y^2z + x^3y^{-6}z^{-2} + 3x^{-2}y^{-1}z^{-2} - 2 - x^{-1}y - y^{-1}z^{-1} - x^2y^{-4}z^{-1} - xy^{-3}z^{-1};$$

it is a K3 surface of geometric Picard rank 6, and the Hodge numbers of $PH^2(\mathcal{X})$ are $(1, 14, 1)$. For $p = 2^{15} - 19$, using our current implementation, in $6m$ $20s$ we obtain the “interesting” factor of $Z(\mathcal{X}, t)$

$$pQ(t/p) = (1 - t)(1 + t)(p + 33305t^1 + 1564t^2 - 14296t^3 - 11865t^4$$

$$+ 5107t^5 + 27955t^6 + 25963t^7 + 27955t^8 + 5107t^9$$

$$- 11865t^{10} - 14296t^{11} + 1564t^{12} + 33305t^{13} + pt^{14}).$$

We present the running times for other $p$ in Table 3, where the peak memory usage was about 144MB.

The vertices of the associated polytope correspond to the first six terms displayed; the remaining terms are interior points. We depict this polytope of volume 8 at right.

We know of no previous algorithm that can compute $Z(\mathcal{X}, t)$ for $p$ in this range. The defining polynomial is “dense” from the point of the Sperber–Voight algorithm [SV13], which is based on Dwork cohomology and scales with the number of monomials away from the vertices of the Newton polytope.

Example 6.4. Let $\mathcal{X}$ be the smooth projective surface in $\mathbb{P}^3$ defined by the fully dense, randomly chosen quartic polynomial

$$-9x^4 - 10x^3y - 9x^2y^2 + 2xy^3 - 7y^4 + 6x^3z + 9x^2yz - 2xy^2z + 3y^3z$$

$$+ 8x^2z^2 + 6y^2z^2 + 2xz^3 + 7yz^3 + 9z^4 + 8x^3w + x^2yw - 8xy^2w$$

$$- 7y^3w + 9x^2zw - 9yzw + 3y^2zw - xz^2w - 3yz^2w + z^3w - x^2w^2$$

$$- 4xyw^2 - 3xz^2w^2 + 8yzw^2 - 6z^2w^2 + 4xw^3 + 3yw^3 + 4zw^3 - 5w^4;$$

then $\Delta$ is the 3-simplex (tetrahedron) of volume $32/3 \approx 10.66$. For this example, we have $PH^2(\mathcal{X}) \simeq H^3(\mathbb{P}^3 \setminus \mathcal{X})$, the Hodge numbers are $(1, 19, 1)$, and

$$Z(\mathcal{X}, t)^{-1} = (1 - t)(1 - pt)(1 - p^2t)Q(t),$$

<table>
<thead>
<tr>
<th>$p$</th>
<th>time</th>
<th>$p$</th>
<th>time</th>
<th>$p$</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^7 - 1$</td>
<td>6.46s</td>
<td>$2^{10} - 3$</td>
<td>18.93s</td>
<td>$2^{13} - 1$</td>
<td>1m 46s</td>
</tr>
<tr>
<td>$2^8 - 5$</td>
<td>9.50s</td>
<td>$2^{11} - 9$</td>
<td>31.34s</td>
<td>$2^{14} - 3$</td>
<td>3m 24s</td>
</tr>
<tr>
<td>$2^9 - 3$</td>
<td>12.64s</td>
<td>$2^{12} - 3$</td>
<td>56.23s</td>
<td>$2^{15} - 19$</td>
<td>6m 20s</td>
</tr>
</tbody>
</table>

Table 3. Running times for Example 6.3.
Table 4. Running times for Example 6.4.

<table>
<thead>
<tr>
<th>$p$</th>
<th>time</th>
<th>$p$</th>
<th>time</th>
<th>$p$</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^7 - 1$</td>
<td>25.41s</td>
<td>$2^{10} - 3$</td>
<td>1m 30s</td>
<td>$2^{13} - 1$</td>
<td>9m 26s</td>
</tr>
<tr>
<td>$2^8 - 17$</td>
<td>37.73s</td>
<td>$2^{11} - 9$</td>
<td>2m 37s</td>
<td>$2^{14} - 3$</td>
<td>18m 42s</td>
</tr>
<tr>
<td>$2^9 - 3$</td>
<td>55.82s</td>
<td>$2^{12} - 3$</td>
<td>4m 50s</td>
<td>$2^{15} - 19$</td>
<td>36m 29s</td>
</tr>
</tbody>
</table>

where $\deg Q(t) = 21$. For $p = 2^{15} - 19$, we obtain

$$pQ(t/p) = (1 + t)(p - 53159r^1 + 10023r^2 - 3204r^3 + 49736r^4 - 56338r^5 + 43086r^6$$

$$- 48180r^7 + 44512r^8 - 42681r^9 + 47794r^{10} - 42681r^{11} + 44512r^{12} - 48180r^{13}$$

$$+ 43086r^{14} - 56338r^{15} + 49736r^{16} - 3204r^{17} + 10023r^{18} - 53159r^{19} + pt^{20})$$

using the controlled AKR algorithm in 38m 27s; our current implementation takes roughly the same time. We present the running times for other $p$ in Table 4. The memory footprint was about 230MB.

Unfortunately, the deformation method is not suitable for dense quartics with $p$ in this range. For example, for $p = 31$ the running time was 2h 8m and its memory footprint was around 7GB, and both time and space should scale linearly with $p$.

### 7. Calabi–Yau threefolds

We next consider Calabi–Yau threefolds. Unlike for K3 surfaces, the middle Betti numbers of Calabi–Yau threefolds are not a priori bounded; the largest value of which we are aware is 984 (found in [KS00]).

A common example is a smooth quintic surface in $\mathbb{P}^4$. Again, additional constructions arise from generic hypersurfaces in weighted projective spaces, of which there are 7,555 in all, or more generally from toric varieties corresponding to reflexive 4-dimensional polytopes, of which there are 473,800,776 in all [KS00].

In all of the following examples, we worked modulo $p^6$ in order to obtain $Q(t)$ and our memory footprint ranged between 100MB and 270MB.

**Example 7.1.** Consider the projective quintic threefold $\mathcal{X} \subset \mathbb{P}^3_{\mathbb{F}_p}$ defined by

$$x_0^5 + x_1^5 + x_2^5 + x_3^5 + x_4^5 + x_0x_1x_2x_3x_4 = 0;$$

it is a member of the Dwork pencil. We have

$$Z(\mathcal{X}, t) = \frac{R_1(pt)^{20}R_2(pt)^{30}Q(t)}{(1-t)(1-pt)(1-p^2t)(1-p^3t)},$$

where $R_1$ and $R_2$ are the numerators of the zeta functions of certain curves given by a formula of Candelas, de la Ossa and Rodriguez Villegas [CdlORV03].

As it is presented, we would work with $\mathbb{P}_\Delta = \mathbb{P}^4$, where $\Delta$ is the 4-simplex of volume 625/24. As in Example 6.1, the monomials of the equation generate a sublattice of index 53 in $\mathbb{Z}^4$, so we may instead
work with a polytope whose volume is smaller by a factor of $5^3$. For $p = 2^{20} - 3$, we compute the “interesting” factor

$$Q(t) = 1 - 1576492860t^1 + 2672053179370pt^2 - 1576492860p^3t^3 + p^6t^4$$

in 11m 18s; if we instead had tried to apply the controlled AKR algorithm to compute $Q(t)$ (and not the other factors) we extrapolate that it would take us at least 120 days. We present the running times for other $p$ in Table 5.

Since this is a “small” perturbation of the Fermat threefold, we again attempted to confirm these results using the deformation method; however, this was again hampered by the fact that the Pancratz–Tuitman implementation works in $\mathbb{P}^3$ instead of $\mathbb{P}^3$. For $p = 7$, it took 5h 4m and its memory footprint was around 12GB.

**Example 7.2.** Let $X$ be the threefold defined by

$$x_0^8 + x_1^5x_2 + x_0^2x_1^2x_2x_3 + x_1x_2^3x_3 + x_1^2x_3^3 + x_0x_1x_2x_3x_4 + x_2x_3x_4^2$$

in the weighted projective space $\mathbb{P}(1, 14, 18, 20, 25)$. The Newton polytope has volume $11/3 \approx 3.67$; by changing the lattice we may instead work with a polytope of volume $1/3 \approx 0.33$. In this setting, the Hodge numbers of $\text{PH}^3(X)$ are $(1, 1, 1, 1)$.

For $p = 2^{20} - 3$, we compute the “interesting” factor of $Z(X, t)$

$$1 - 618297672t^1 + 390956360946pt^2 - 618297672p^3t^3 + p^6t^4$$

in 32m 33s. We present the running times for other $p$ in Table 6.
Example 7.3. Let $X$ be the threefold defined by
\[ x_1^7 + x_0^5 x_1 x_2 + x_0^2 x_1^2 x_2 x_3 + x_0^4 x_2 x_4 + x_0 x_2^3 x_3 + x_0^2 x_3^2 + x_0 x_1 x_2 x_3 x_4 + x_2 x_3 x_4^2 \]

in the weighted projective space $\mathbb{P}(10, 11, 16, 19, 21)$. Again, by choosing the right lattice, we reduce the volume of the Newton polytope from $55/12 \approx 4.58$ to $11/24 \approx 0.46$, and the Hodge numbers of $PH^3(X)$ are $(1, 2, 2, 1)$. For $p = 2^{20} - 3$, we computed the “interesting” factor of $Z(X, t)$

\[
1 - 2068001468t^1 + 3449674041773pt^2 - 3772715295733197p^2t^3 + 3449674041773p^4t^4 - 2068001468p^6t^5 + p^9 t^6
\]

in 2h 10m. We present the running times for other $p$ in Table 7.

Example 7.4. Let $X$ be the closure in $\mathbb{P}_\Delta$ (which is not a weighted projective space) of the threefold defined by the Laurent polynomial

\[ xyz^2 w^3 + x + y + z - 1 + y^{-1}z^{-1}x^{-2}y^{-1}z^{-2}w^{-3} = 0. \]

Choosing the correct lattice reduces the volume of the Newton polytope from $9/8 \approx 1.12$ to $3/8 \approx 0.38$, and the Hodge numbers of $PH^3(X)$ are $(1, 2, 2, 1)$. For $p = 2^{20} - 3$, we computed the “interesting” factor of $Z(X, t)$

\[
(1 + 718pt + p^3 t^2)(1 + 1188466826t^1 + 1915150034310pt^2 + 1188466826p^3 t^3 + p^6 t^4)
\]

in 1h 15m. We present the running times for other $p$ in Table 8.
8. Cubic fourfolds

For our final example, we consider a cubic fourfold. For \( X \) a smooth cubic fourfold in \( \mathbb{P}^5 \), \( \dim H^4(X) = 23 \) and the Hodge numbers are \((0, 1, 21, 1, 0)\).

In this example, we worked modulo \( p^6 \) in order to obtain \( Q(t) \).

**Example 8.1.** Let \( \mathcal{X} \) be the smooth projective cubic fourfold in \( \mathbb{P}^5_{\mathbb{F}_p} \) defined by

\[
x_0^3 + x_1^3 + x_2^3 + (x_0 + x_1 + 2x_2)^3 + x_3^3 + x_4^3 + 2(x_0 + x_3)^3 + 3(x_1 + x_4)^3 + (x_2 + x_5)^3;
\]

it is nondegenerate in \( \mathbb{P}^5 \). For \( p = 31 \), in 21h 31m we computed

\[
Z(\mathcal{X}, t)^{-1} = (1 - t)(1 - pt)(1 - p^2t)(1 - p^3t)(1 - p^4t) Q(t),
\]

where the “interesting” factor is an irreducible Weil polynomial given by

\[
p Q(t/p^2) = p - 7t^1 + 21t^2 - 52t^3 - 8t^4 - 28t^5 + 21t^6 + 35t^7 + 39t^9 + 62t^{10} + 23t^{11}
\]

\[
+ 62t^{12} + 39t^{13} + 35t^{15} + 21t^{16} - 28t^{17} - 8t^{18} - 52t^{19} + 21t^{20} - 7t^{21} + pt^{22};
\]

the coefficient of \( t^1 \) may be confirmed independently by counting \( \mathcal{X}(\mathbb{F}_p) \) using the Sage function `count_points`. For \( p = 127 \) the running time was 23h 15m and for \( p = 499 \) it was 24h 55m; in both cases, the “interesting” factor is an irreducible Weil polynomial. In these computations, the memory footprint was around 36.5GB.

In dimension 4, the bottleneck seems to be the linear algebra required to set up controlled reduction. For \( p = 31 \), more than half of the running time (15h 32m) is spent solving a linear problem of size \( 15,504 \times 37,128 \) modulo \( p^6 \). With careful handling of this step (e.g., avoiding Hensel lifts) we would expect a significant speedup.

Note that the defining equation for \( \mathcal{X} \) is quite sparse. To assess the effect of this sparsity, as well as to cross-check the answer, we recomputed \( Z(\mathcal{X}, t) \) after applying a random linear change of variables to obtain a dense defining equation. For \( p = 31 \), in 27h 55m and using about 41GB we obtained the same value for \( Z(\mathcal{X}, t) \) as above.

Recall from the Introduction that a cubic fourfold is coplanar if it is defined by an expression \( \sum_{i=1}^{10} a_i^3 \) in which each \( a_i \) is a linear form and some four of the \( a_i \) are linearly dependent. Ranestad and Voisin [RV17] showed that the Zariski closure \( D_{\text{copl}} \) of the coplanar locus is a divisor on the moduli space of cubic fourfolds. Example 8.1 is a nonspecial coplanar cubic fourfold: the existence of a primitive codimension-2 cycle class would imply that \( p Q(t/p^2) \) has a cyclotomic factor. This shows (modulo detailed analysis of the algorithm) that \( D_{\text{copl}} \) is not a Noether–Lefschetz divisor.

**References**


While it is not needed here, the Tate conjecture for ordinary cubic fourfolds is known [Lev01].


Paul Monsky, *p-adic analysis and zeta functions*, Lectures in Mathematics, Department of Mathematics, Kyoto University, no. 4, Kinokuniya, Tokyo, 1970. MR 0282981


Jan Tuitman, *Counting points on curves using a map to \(\mathbb{P}^1\)*, Math. Comp. 85 (2016), no. 298, 961–981. MR 3434890

Jan Tuitman, *Counting points on curves using a map to \(\mathbb{P}^1\), II*, Finite Fields Appl. 45 (2017), 301–322. MR 3631366


Received 1 Mar 2018. Revised 24 Aug 2018.

EDGAR COSTA: edgarc@mit.edu

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, United States

DAVID HARVEY: d.harvey@unsw.edu.au

School of Mathematics and Statistics, University of New South Wales, Sydney, Australia

KIRAN S. KEDLAYA: kedlaya@ucsd.edu

Department of Mathematics, University of California, San Diego, La Jolla, CA, United States
The Algorithmic Number Theory Symposium (ANTS), held biennially since 1994, is the premier international forum for research in computational number theory. ANTS is devoted to algorithmic aspects of number theory, including elementary, algebraic, and analytic number theory, the geometry of numbers, arithmetic algebraic geometry, the theory of finite fields, and cryptography.

This volume is the proceedings of the thirteenth ANTS meeting, held July 16-20, 2018, at the University of Wisconsin-Madison. It includes revised and edited versions of 28 refereed papers presented at the conference.

Edited by Renate Scheidler and Jonathan Sorenson

CONTRIBUTORS

Simon Abelard
Sonny Arora
Vishal Arul
Angelica Babei
Jens-Dietrich Bauch
Alex J. Best
Jean-François Biasse
Alin Bostan
Reinier Bröker
Nils Bruin
Xavier Caruso
Stephanie Chan
Qi Cheng
Gilles Christol
Owen Colman
Edgar Costa
Philippe Dumas
Kirsten Eisenträger
Claus Fieker
Shuhong Gao
Pierrick Gaudry
Alexandre Gélin
Alexandru Ghitza
Laurent Grémy
Jeroen Hanselman
David Harvey
Tommy Hofmann
Everett W. Howe
David Hubbard
Kiran S. Kedlaya
Thorsten Kleinjung
David Kohel
Wanlin Li
Richard Magner
Anna Medvedovsky
Michael Musty
Ha Thanh Nguyen Tran
Christophe Ritzenthaler
David Roe
J. Maurice Rojas
Nathan C. Ryan
Renate Scheidler
Sam Schiavone
Andrew Shallue
Jeroen Sijsling
Carlo Sircana
Jonathan Sorenson
Pierre-Jean Spaenlehauer
Andrew V. Sutherland
Nicholas Triantafillou
Joris van der Hoeven
Christine Van Vredendaal
John Voight
Daqing Wan
Lawrence C. Washington
Jonathan Webster
Benjamin Wesolowski
Yinan Zhang
Alexandre Zotine