V. Golyshev and A. Mellit

Computing Frobenius traces in a non-hypergeometric family

Given a family of algebraic varieties X/S defined over $\mathbb{Z}$ and a meromorphic relative differential form ω of degree d its periods satisfy a Picard-Fuchs differential equation. For a prime number p consider $a_p(s)$, the trace of the Frobenius operator acting on $H^d(X_s)$.

Problem:

Compute $a_p(s)$ looking only at the differential equation (+ interesting formal solutions).

The test case

Reasons:

- A DE where we have a geometric realization, so we have an alternative way to compute $a_p(s)$ and check results.

- Not a hypergeometric case, non-rigid.

Apéry's differential equation

Family of elliptic curves with $\Gamma_1(5)$-level structure:

$$\Lambda(x,y) = t^{-1}, \qquad t \neq 0,\ t^2 + 11t - 1 \neq 0;$$

$$\Lambda(x,y) = x^{-1}y^{-1}(1+x)(1+y)(1+x+y).$$

Relative differential form ω can be found from

$$\frac{dx}{x} \wedge \frac{dy}{y} = \omega \wedge \frac{d\Lambda}{\Lambda}.$$

Picard-Fuchs differential equation (Beukers, Zagier):

$$(D^2 - t(11D^2 + 11D + 3) - t^2(D+1)^2)\varphi = 0,$$

$$D = t\frac{d}{dt}.$$

Period integral:

$$(2\pi i)^{-1} \int_{S^1 \times S^1} \frac{1}{1 - t\Lambda} \frac{dx}{x} \wedge \frac{dy}{y} \;=\; 2\pi i \sum_{n=0}^{\infty} b_n t^n.$$

Formal solution:

$$\varphi(t) = \sum_{n=0}^{\infty} b_n t^n = 1 + 3t + 19t^2 + 147t^3 + 1251t^4 + \cdots .$$

Sequence b_n is a solution to the corresponding recurrence equation.

Use b_n to compute $a_p(t)$.

Traces obtained by counting points

$p = 7$

```
>>> [count_frob(k, 7) for k in range(1, 7)]
[-2, -2, -2, 3, 3, -2]
```

$p = 17$

```
>>> [count_frob(k, 17) for k in range(1, 17)]
[-2, 3, 3, -2, -2, -2, -7, 3, 3, -2, 3, -7, 8, -2, 3, -2]
```

$p = 19$

```
>>> [count_frob(k, 19) for k in range(1, 19)]
[0, 5, 0, -5, 0, 0, 0, 0, 5, -1, 5, 5, -5, -5, 0, -5, -1, 0]
```

Approaches

- Dwork's approach
- Dwork + Fourier transform
- Stienstra's approach
- Stienstra + Mellin transform

Dwork

Fix m. Look for a rational function $R(t)$ whose Taylor expansion agrees with $\frac{\varphi(t)}{\varphi(t^p)}$ modulo p^m (continued fractions). For a non-supersingular value of t

$$a_p(t) \equiv R(\tau(t)) + \frac{p}{R(\tau(t))} \mod p^m.$$

First approximation:

$$a_p(t) \equiv \sum_{n=0}^{p-1} b_n t^n \mod p$$

```
>>> [sum(B(n) * t**n for n in range(17))%17 for t in range(1, 17)]
[15, 3, 3, 15, 15, 15, 10, 3, 3, 15, 3, 10, 8, 15, 3, 15]
>>> [sum(B(n) * t**n for n in range(19))%19 for t in range(1, 19)]
[0, 5, 0, 14, 0, 0, 0, 0, 5, 18, 5, 5, 14, 14, 0, 14, 18, 0]
```

Higher approximations:

```
>>> set_precision(10*19**2, 4); p=19
>>> phi2=phi(B)/pari.subst(phi(B), X, X**p)
>>> res = pari.lift(pari.Pol(phi2*pari.Mod(1, p)))
>>> res += p * pari.lift(guess_rational((phi2-res)/p*pari.Mod(1, p)))
Cutoff: 2894 346
>>> e=[pari.subst(res, X, teichmuller(k, p)) for k in range(p)]
>>> e[4]+p/e[4]
14 + 18*19 + 2*19^2 + O(19^4)
>>> e[2]+p/e[2]
5 + 6*19^2 + 11*19^3 + O(19^4)
>>> e[10] # singular
18 + 18*19 + 17*19^2 + 7*19^3 + O(19^4)
```

```
>>> set_precision(4*7**3, 4); p=7
>>> phi2=phi(B)/pari.subst(phi(B), X, X**p)
>>> res = pari.lift(pari.Pol(phi2*pari.Mod(1, p)))
>>> res += p * pari.lift(guess_rational((phi2-res)/p*pari.Mod(1, p)))
Cutoff: 1280 39
>>> res += p**2 * pari.lift(guess_rational((phi2-res)/p**2*pari.Mod(1, p)))
Cutoff: 610 342
>>> e=[pari.subst(res, X, teichmuller(k, p)) for k in range(p)]
>>> e[1]+p/e[1]
5 + 6*7 + 6*7^2 + 5*7^3 + O(7^4)
>>> e[4]+p/e[4]
3 + 4*7^3 + O(7^4)
```

Fourier transform

Notation:

$$\pi^{p-1} = -p,\ b_n^! = \frac{b_n}{n!},\ \varphi^!(t) = \sum b_n^! t^n.$$

Pass to

$$\psi(t) = \frac{\varphi^!(\pi t)}{\varphi^!(\pi t^p)}.$$

Now ψ converges in the closed unit disk and computes the first eigenvalue of the Fourier transform. The rank of the DE satisfied by $\varphi^!$ is 3. To obtain the other two eigenvalues we use a trick.

Exterior powers

Compute $\wedge^2$ and $\wedge^3$ of the DE satisfied by $\varphi^!$, their solutions and ψ.

To find solutions of exterior products we deform the sequence $b^!_n$ to obtain a sequence $b^!_{n+\varepsilon}$ which satisfies the same recursion as $b^!_n$, but with n replaced by $n+\varepsilon$ ($\varepsilon^3 = 0$):

$$b^!_{n+\varepsilon} : 1,\ 3 + 2\varepsilon - 4\varepsilon^2,\ \tfrac{19}{2} + \tfrac{9}{2}\varepsilon - 14\varepsilon^2, \ldots$$

Solutions to the DE of $\varphi^!$ are given by (note the logs!)

$$s_i(t) = \frac{\partial^i}{i!\partial\varepsilon^i}\bigg|_{\varepsilon=0} \sum_n b^!_{n+\varepsilon} t^{n+\varepsilon} \quad i = 0, 1, 2.$$

The Wronskians of (s_0), $(s_0,\ s_1)$, $(s_0,\ s_1,\ s_2)$ are solutions to the exterior power DEs. Put

$$\varphi^!_i = \frac{\partial^i}{i!\partial\varepsilon^i}\bigg|_{\varepsilon=0} \sum_n b^!_{n+\varepsilon} t^n \quad i = 0, 1, 2.$$

Then

$$\varphi^!_{\wedge^2} = \varphi^{!2}_0 + \varphi^!_0 D\varphi^!_1 - \varphi^!_1 D\varphi^!_0,$$

$$\varphi^!_{\wedge^3} = \det\begin{pmatrix} \varphi^!_0 & \varphi^!_1 & \varphi^!_2 \\ D\varphi^!_0 & D\varphi^!_1 + \varphi^!_0 & D\varphi^!_2 + \varphi^!_1 \\ D^2\varphi^!_0 & D^2\varphi^!_1 + 2D\varphi^!_0 & D^2\varphi^!_2 + 2D\varphi^!_1 + \varphi^!_0 \end{pmatrix}.$$

In fact, in our example $\varphi^!_{\wedge^3}(t) = \exp(11t)$.

We obtain $\psi_{\wedge^2}$, $\psi_{\wedge^3}$ from $\varphi^!_{\wedge^2}$, $\varphi^!_{\wedge^3}$ in the same way as ψ.

Now for $k = 0, \ldots, p-1$ we recover the eigenvalues of the Fourier transform as:

$$\psi(\tau(k)), \quad p\frac{\psi_{\wedge^2}(\tau(k))}{\psi(\tau(k))}, \quad p^2\frac{\psi_{\wedge^3}(\tau(k))}{\psi_{\wedge^2}(\tau(k))}.$$

The trace

$$a_p^!(k) = \psi(\tau(k)) + p\frac{\psi_{\wedge^2}(\tau(k))}{\psi(\tau(k))} + p^2\frac{\psi_{\wedge^3}(\tau(k))}{\psi_{\wedge^2}(\tau(k))}$$

and we recover $a_p(t)$ by taking the inverse Fourier transform:

$$a_p(t) = -\frac{1}{p}\sum_{k=0}^{p-1} a_p^!(k)\zeta^{kt} + p + 1,$$

where $\zeta^p = 1$, $\zeta \equiv 1 + \pi \mod \pi^2$, $\tau(x)$ denotes the Teichmüller representative of x.

```
>>> w1=phi(fact(B))
>>> w2=phi_w2(fact_eps(B_eps(2), 2))
>>> w3=phi_w3(fact_eps(B_eps(3), 3))
>>> p=17
>>> psi1=psi(w1, p)
>>> psi2=psi(w2, p)
>>> psi3=psi(w3, p)
>>> zz=zeta(p)
>>> e1=[evaluate(psi1, teichmuller(k, p)) for k in range(p)]
>>> e2=[evaluate(psi2, teichmuller(k, p)) for k in range(p)]
>>> e3=[evaluate(psi3, teichmuller(k, p)) for k in range(p)]
>>> tr_f=[e1[k] + p*e2[k]/e1[k] + p**2*e3[k]/e2[k] for k in range(p)]
>>> tr=[-sum(tr_f[k]*zz**(k*t) for k in range(p))/p + p + 1 \
... for t in range(p)]
>>> pari.lift(tr[1])[0]
15 + 16*17 + 7*17^2 + O(17^3)
```

```
>>> p=19
>>> psi1=psi(w1, p)
>>> psi2=psi(w2, p)
>>> psi3=psi(w3, p)
>>> zz=zeta(p)
>>> e1=[evaluate(psi1, teichmuller(k, p)) for k in range(p)]
>>> e2=[evaluate(psi2, teichmuller(k, p)) for k in range(p)]
>>> e3=[evaluate(psi3, teichmuller(k, p)) for k in range(p)]
>>> tr_f=[e1[k] + p*e2[k]/e1[k] + p**2*e3[k]/e2[k] for k in range(p)]
>>> tr=[-sum(tr_f[k]*zz**(k*t) for k in range(p))/p + p + 1 \
... for t in range(p)]
>>> pari.lift(tr[1])[0]
12*19^2 + O(19^3)
>>> pari.lift(tr[2])[0]
5 + 9*19^2 + O(19^3)
```

```
>>> p=7
>>> psi1=psi(w1, p)
>>> psi2=psi(w2, p)
>>> psi3=psi(w3, p)
>>> zz=zeta(p)
>>> e1=[evaluate(psi1, teichmuller(k, p)) for k in range(p)]
>>> e2=[evaluate(psi2, teichmuller(k, p)) for k in range(p)]
>>> e3=[evaluate(psi3, teichmuller(k, p)) for k in range(p)]
>>> tr_f=[e1[k] + p*e2[k]/e1[k] + p**2*e3[k]/e2[k] for k in range(p)]
>>> tr=[-sum(tr_f[k]*zz**(k*t) for k in range(p))/p + p + 1 \
... for t in range(p)]
>>> pari.lift(tr[1])[0]
5 + 6*7 + 6*7^2 + O(7^3)
>>> pari.lift(tr[4])[0]
3 + O(7^3)
```

Stienstra's theorem

For a Laurent polynomial Λ with the only interior point of Newton polytope 0 consider the sequence of constant terms:

$$b_{\Lambda,n} = c.t.(\Lambda^n).$$

Then

$$\sum_{i=0}^{d} c_i b_{\Lambda,p^{n-i}-1} \equiv 0 \mod p^n,$$

where $\sum_{i=0}^{d} c_i x^i = \det(1 - x \cdot \mathrm{Frob})$ acting on the middle cohomology of the hypersurface defined by equation $\Lambda = 0$.

Fix $k = 1, \ldots, p-1$. To obtain input to the Stienstra's theorem we need to study the sequence of constant terms:

$$b_{n,t_0} = c.t.(\Lambda t_0 - 1)^n = \sum_{k=0}^{n} \binom{n}{k} (-1)^{n-k} b_k t_0^k.$$

The corresponding generating function is

$$\varphi_{t_0}(t) = \sum_{n=0}^{\infty} b_{n,t_0} t^n = \frac{1}{1+t} \varphi\left(\frac{t_0 t}{1+t}\right).$$

In the non-supersingular case the first eigenvalue is:

$$e_p(t_0) = \lim_{n\to\infty} \frac{b_{p^n-1,t_0}}{b_{p^{n-1}-1,t_0}} \qquad \text{(p-adically)}.$$

$$a_p(t_0) = e_p(t_0) + \frac{p}{e_p(t_0)}.$$

```
>>> p=19
>>> phi_t=[pari.subst(phi(B), X, t*X/(1+X))/(1+X) for t in range(p)]
>>> e=[padic(phi_t[k][p**2-1]/phi_t[k][p-1], p) for k in range(p)]
>>> e[2]+p/e[2]
5 + 2*19^2 + 19^3 + O(19^4)
>>> e[4]+p/e[4]
14 + 18*19 + 8*19^2 + 8*19^3 + O(19^4)
>>> e[10] # singular
18 + 18*19 + 13*19^2 + 7*19^3 + O(19^4)
>>> e[1]+p/e[1] # supersingular
6 + 5*19 + 13*19^2 + 13*19^3 + O(19^4)
```

Mellin transform

Use b_n without the shift.

Fourier tranform: DE to DE, φ to $\varphi^!$.

Mellin transform: DE to recursion equation, φ to (b_n).

For $k = 1, \ldots, p-1$ the information about the Mellin transform at k is captured by the subsequence

$$b_{k,n} = b_{k\frac{p^n-1}{p-1}},$$

namely

$$\sum_{i=0}^{d} b_{k,n-i}c_i \longrightarrow 0,$$

where $\sum c_i x^i = \det(1 - x \cdot \mathrm{Frob})$.

Exterior powers

Since the length of the recursion (degree in t) is 2, we should have two eigenvalues of the Mellin transform. To obtain all necessary information we look at the second exterior power of the recursion. Let b'_n be another solution to the same recursion as b_n, but starting with $0, 1$. Then

$$b_{\wedge^2,n} = b_n b'_{n+1} - b_{n+1} b'_n = \frac{(-1)^n}{(n+1)^2}.$$

Hence the Frobenius action on $\wedge^2$ (with the right Tate twist) becomes $(-1)^k p^2$ for $k \neq p-1$.

We recover the trace as (m stands for "Mellin")

$$a_p^m(k) = \lim_{n\to\infty} \frac{b_{k,n} + (-1)^k p^2 b_{k,n-2}}{b_{k,n-1}}.$$

For $k = p-1$ we use $a^m_{p,p-1} = 1$. Then, taking the inverse Mellin transform we obtain

$$a_p(t) = \frac{1}{1-p}\left(1 + \sum_{k=1}^{p-1} a_p^m(k)\tau(t)^k\right).$$

```
>>> p=19; p1=(p**4-1)//(p-1); p2=(p**3-1)//(p-1); p3=(p**2-1)//(p-1)
>>> w2=[p**2 * padic((-1)**k, p) if k!=p-1 else 0 for k in range(p)]
>>> tr_m=[padic((B(p1*k) + w2[k]*B(p3*k))/B(p2*k), p) for k in range(p)]
>>> tr=[(1+sum(tr_m[k]*teichmuller(t, p)**k for k in range(1,p)))/(1-p) \
... for t in range(p)]
>>> tr[1]
O(19^4)
>>> tr[2]
5 + O(19^4)
>>> tr[4]
14 + 18*19 + 18*19^2 + 18*19^3 + O(19^4)
>>> tr[10]
18 + 18*19 + 18*19^2 + 18*19^3 + O(19^4)
```

```
>>> p=17; p1=(p**4-1)//(p-1); p2=(p**3-1)//(p-1); p3=(p**2-1)//(p-1)
>>> w2=[p**2 * padic((-1)**k, p) if k!=p-1 else 0 for k in range(p)]
>>> tr_m=[padic((B(p1*k) + w2[k]*B(p3*k))/B(p2*k), p) for k in range(p)]
>>> tr=[(1+sum(tr_m[k]*teichmuller(t, p)**k for k in range(1,p)))/(1-p) \
... for t in range(p)]
>>> tr[1]
15 + 16*17 + 16*17^2 + 16*17^3 + O(17^4)
>>> tr[2]
3 + O(17^4)
>>> tr[7]
10 + 16*17 + 16*17^2 + 16*17^3 + O(17^4)
>>> tr[13]
8 + O(17^4)
```

```
>>> p=7; p1=(p**4-1)//(p-1); p2=(p**3-1)//(p-1); p3=(p**2-1)//(p-1)
>>> w2=[p**2 * padic((-1)**k, p) if k!=p-1 else 0 for k in range(p)]
>>> tr_m=[padic((B(p1*k) + w2[k]*B(p3*k))/B(p2*k), p) for k in range(p)]
>>> tr=[(1+sum(tr_m[k]*teichmuller(t, p)**k for k in range(1,p)))/(1-p) \
... for t in range(p)]
>>> tr[1]
5 + 6*7 + 6*7^2 + 6*7^3 + O(7^4)
>>> tr[2]
5 + 6*7 + 6*7^2 + 6*7^3 + O(7^4)
>>> tr[3]
5 + 6*7 + 6*7^2 + 6*7^3 + O(7^4)
>>> tr[4]
3 + O(7^4)
```