Computing L-functions of hyperelliptic curves

Andrew V. Sutherland

Massachusetts Institute of Technology

ICTP Workshop on the Arithmetic of Hyperelliptic Curves
September 8, 2017

Joint with (various subsets of) A. Booker, D. Platt, D. Harvey, M. Massierer.

Zeta functions and L-functions

Let $X/\mathbb{Q}$ be a nice (smooth, projective, geometrically integral) curve of genus g. For primes p of good reduction (for X) we have a zeta function

$$Z(X_p; s) := \exp\left(\sum_{r\geq 1} \#X_p(\mathbb{F}_{p^r})\frac{T^r}{r}\right) = \frac{L_p(T)}{(1-T)(1-pT)},$$

in which the L-polynomial $L_p \in \mathbb{Z}[T]$ in the numerator satisfies

$$L_p(T) = T^{2g}\chi_p(1/T) = 1 - a_pT + \cdots + p^gT^{2g};$$

here $\chi_p(T)$ is the charpoly of the Frobenius endomorphism of $\mathrm{Jac}(X_p)$ (this implies $\#\mathrm{Jac}(X_p) = L_p(1)$, for example). The L-function of X is

$$L(X,s) = L(\mathrm{Jac}(X), s) := \sum_{n\geq 1} a_n n^{-s} := \prod_p L_p(p^{-s})^{-1},$$

where the Dirichlet coefficients $a_n \in \mathbb{Z}$ are determined by the $L_p(T)$. In particular, $a_p = p + 1 - \#X_p(\mathbb{F}_p)$ is the trace of Frobenius.

The Selberg class with polynomial Euler factors

The Selberg class S^{poly} consists of Dirichlet series $L(s) = \sum_{n\geq 1} a_n n^{-s}$:

1. $L(s)$ has an analytic continuation that is holomorphic at $s \neq 1$;
2. For some $\gamma(s) = Q^s \prod_{i=1}^{r} \Gamma(\lambda_i s + \mu_i)$ and ε, the completed L-function $\Lambda(s) := \gamma(s)L(s)$ satisfies the functional equation

$$\Lambda(s) = \varepsilon\overline{\Lambda(1-\bar{s})},$$

   where $Q > 0$, $\lambda_i > 0$, $\mathrm{Re}(\mu_i) \geq 0$, $|\varepsilon| = 1$. Define $\deg L := 2\sum_i^r \lambda_i$.
3. $a_1 = 1$ and $a_n = O(n^{\epsilon})$ for all $\epsilon > 0$ (Ramanujan conjecture).
4. $L(s)$ has an Euler product $L(s) = \prod_p L_p(p^{-s})^{-1}$ in which each local factor $L_p \in \mathbb{Z}[T]$ has degree at most $\deg L$.

The Dirichlet series $L_{\text{an}}(s, X) := L(X, s + \frac{1}{2})$ satisfies (3) and (4), and conjecturally lies in S^{poly}; for $g = 1$ this is known (via modularity).

Strong multiplicity one

Theorem (Kaczorowski-Perelli 2001)

If $A(s) = \sum_{n\geq 1} a_n n^{-s}$ *and* $B(s) = \sum_{n\geq 1} b_n n^{-s}$ *lie in* S^{poly} *and* $a_p = b_p$ *for all but finitely many primes* p*, then* $A(s) = B(s)$*.*

Corollary

If $L_{\mathrm{an}}(s,X)$ *lies in* S^{poly} *then it is determined by (any choice of) all but finitely many coefficients* a_p*. In particular, all of the local factors are completely determined by the Frobenius traces* a_p *at good primes.*

Henceforth we assume that $L_{\mathrm{an}}(s,X) \in S^{\mathrm{poly}}$.

Let $\Gamma_{\mathbb{C}}(s) := 2(2\pi)^s\Gamma(s)$, and define $\Lambda(X,s) := \Gamma_{\mathbb{C}}(s)^g L(X,s)$. Then

$$\Lambda(X,s) = \varepsilon N^{1-s}\Lambda(X,2-s).$$

where the analytic root number $\varepsilon = \pm 1$ and analytic conductor $N \in \mathbb{Z}_{\geq 1}$ are also determined by the Frobenius traces a_p at good primes.

Effective strong multiplicity one

Fix a finite set of primes $\mathcal{S}$ (e.g. bad primes) and an integer M that we know is a multiple of the conductor N (e.g. $M = \Delta(X)$).

There is a finite set of possibilities for $\varepsilon = \pm 1$, $N|M$, and the Euler factors $L_p \in \mathbb{Z}[T]$ for $p \in \mathcal{S}$ (the coefficients of $L_p(T)$ are bounded).

Suppose we know the a_n for all $n \leq c_1\sqrt{M}$ with $p \nmid n$ for $p \in \mathcal{S}$. For a suitably large c_1, exactly one choice of ε, N, and $L_p(T)$ for $p \in \mathcal{S}$ will make it possible for $L(X, s)$ to satisfy its functional equation.[1]

One can explicitly determine a set of $O(N^\epsilon)$ candidate values of c_1, one of which is guaranteed to work; in practice the first one usually works.

This gives an effective algorithm to compute ε, N, and $L_p(T)$ for $p \in \mathcal{S}$, provided we can compute $L_p(T)$ at good $p \leq B$, where $B = O(\sqrt{N})$.

[1] Subject to our assumption that $L_{\text{an}} \in \mathcal{S}^{\text{poly}}$. But if the algorithm fails we have an explicit counterexample to the conjectured Langlands correspondence.

Algorithms to compute zeta functions

Given $X/\mathbb{Q}$ of genus g, we want to compute $L_p(T)$ for all good $p \le B$.

	complexity per prime (ignoring $(\log\log p)^{O(1)}$ factors)		
algorithm	$g = 1$	$g = 2$	$g = 3$
point enumeration	$p \log p$	$p^2 \log p$	$p^3 (\log p)^2$
group computation	$p^{1/4} \log p$	$p^{3/4} \log p$	$p (\log p)^2$
p-adic cohomology	$p^{1/2} (\log p)^2$	$p^{1/2} (\log p)^2$	$p^{1/2} (\log p)^2$
CRT (Schoof-Pila)	$(\log p)^5$	$(\log p)^8$	$(\log p)^{12?}$
average poly-time	$(\log p)^4$	$(\log p)^4$	$(\log p)^4$

For $L(X,s) = \sum a_n n^{-s}$, we only need a_{p^2} for $p^2 \le B$, and a_{p^3} for $p^3 \le B$.
For $1 < r \le g$ we can easily compute a_{p^r} for $p^r \le B$ in time $O(B \log B)$.

Bottom line: it all comes down to computing Frobenius traces.

Warmup: average polynomial-time in genus 1

Let $X : y^2 = f(x)$ with $\deg f = 3, 4$ and $f(0) \neq 0$, and let f_k^n denote the coefficient of x^k in f^n. For each good prime p we have

$$\begin{aligned}
\#X_p(\mathbb{F}_p) &= \sum_{a,b\in\mathbb{F}_p} \left[b^2 = f(a)\right] + N_\infty \\
&\equiv \sum_{a,b\in\mathbb{F}_p} \left(1 - (b^2 - f(a))^{p-1}\right) + N_\infty \\
&\equiv -\sum_{a,b\in\mathbb{F}_p} \sum_r \binom{p-1}{r} b^{2r}(-f(a))^{p-1-r} + N_\infty \\
&\equiv \binom{p-1}{\frac{p-1}{2}} (-1)^{\frac{p-1}{2}} \sum_{a\in\mathbb{F}_p} f(a)^{\frac{p-1}{2}} + N_\infty \\
&\equiv \left(-f_{2p-2}^{(p-1)/2} - f_{p-1}^{(p-1)/2}\right) + \left(1 + (f_4)^{(p-1)/2)}\right) \\
&\equiv 1 - f_{p-1}^{(p-1)/2} \equiv 1 - a_p
\end{aligned}$$

Thus $a_p \equiv f_{p-1}^{(p-1)/2}$. This determines $a_p \in \mathbb{Z}$ for $p \geq 17$, since $|a_p| \leq 2\sqrt{p}$.

Warmup: average polynomial-time in genus 1

We want to compute $f_{p-1}^{(p-1)/2}$ modulo p for many primes p.

The relations $f^{n+1} = f \cdot f^n$ and $(f^{n+1})' = (n+1)f' \cdot f^n$ yield the identity

$$kf_0 f_k^n = \sum_{1 \le i \le d} (i(n+1) - k) f_i f_{k-i}^n,$$

valid for all $k, n \ge 0$. For $d = 3$ (and similarly for $d = 4$), we define

$$v_k^n := [f_{k-2}^n, f_{k-1}^n, f_k^n], \qquad M_k^n := \begin{bmatrix} 0 & 0 & (3n+3-k)f_3 \\ kf_0 & 0 & (2n+2-k)f_2 \\ 0 & kf_0 & (n+1-k)f_1 \end{bmatrix}.$$

For all positive integers k and n we then have

$$v_k^n = \frac{1}{kf_0} v_{k-1}^n M_k^n = \frac{1}{(f_0)^k k!} v_0^n M_1^n \cdots M_k^n.$$

Warmup: average polynomial-time in genus 1

We want to compute $a_p \equiv f_{2n}^n \bmod p$ with $n := (p-1)/2$.
This is the last entry of the vector

$$v_{2n}^n = \frac{1}{f_0^{2n}(2n!)} v_0^n M_1^n \cdots M_{2n}^n = -v_0^n M_1^n \cdots M_{2n}^n$$

reduced modulo $p = 2n+1$.

Observe that $2(n+1) \equiv 1 \bmod p$, so $2M_k^n \equiv M_k \bmod p$, where

$$M_k := \begin{bmatrix} 0 & 0 & (3-2k)f_3 \\ kf_0 & 0 & (2-2k)f_2 \\ 0 & kf_0 & (1-2k)f_1 \end{bmatrix} \in \mathbb{Z}^{3\times 3}$$

is independent of $\boldsymbol{p}$. For each odd prime $p = 2n+1$ we have

$$v_{2n}^n \equiv -\left(\frac{f_0}{p}\right) V_0 M_1 \cdots M_{2n-1} \bmod p \qquad \text{(where } V_0 = [0,0,1]\text{)}.$$

Accumulating remainder tree

Given matrices $M_0, \ldots, M_{n-1}$ and moduli $m_1, \ldots, m_n$, to compute

$$\begin{aligned} M_0 &\bmod m_1 \\ M_0 M_1 &\bmod m_2 \\ M_0 M_1 M_2 &\bmod m_3 \\ M_0 M_1 M_2 M_3 &\bmod m_4 \\ &\cdots \\ M_0 M_1 \cdots M_{n-2} M_{n-1} &\bmod m_n \end{aligned}$$

multiply adjacent pairs and recursively compute

$$\begin{aligned} (M_0 M_1) &\bmod m_2 m_3 \\ (M_0 M_1)(M_2 M_3) &\bmod m_4 m_5 \\ &\cdots \\ (M_0 M_1) \cdots (M_{n-2} M_{n-1}) &\bmod m_n \end{aligned}$$

and adjust the results as required (for better results, use a forest).

Complexity analysis

Assume $\log|f_i| = O(\log B)$. The recursion has depth $O(\log B)$, and in each recursive step we multiply and reduce a bunch of 3×3 matrices with integer entries whose total bitsize is $O(B \log B)$.

We can do all the multiplications/reductions at any given level of the recursion using $O(\mathsf{M}(B \log B)) = B(\log B)^{2+o(1)}$ bit operations.

Total complexity is $B(\log B)^{3+o(1)}$, or $(\log p)^{4+o(1)}$ per prime $p \leq B$.

For a single prime p we do not have a polynomial-time algorithm, but we can give an $O(p^{1/2}(\log p)^{1+o(1)})$ algorithm using the same matrices.

This is a silly way to compute a single a_p in genus 1, but its generalization to genus 2 is competitive, and in genus 3 it yields the fastest practical method known (within the feasible range of p).

Efficiently handling a single prime

Simply computing $V_0 M_1 \cdots M_{p-1}$ modulo p is surprisingly quick (faster than semi-naïve point-counting); it takes $p(\log p)^{1+o(1)}$ time.
For small p (into the thousands), this is the the fastest approach.
But we can do better.

Viewing $M_k \bmod p$ as $M \in \mathbb{F}_p[k]^{3\times 3}$, we compute

$$A(k) := M(k)M(k+1)\cdots M(k+r-1) \in \mathbb{F}_p[k]^{3\times 3}$$

with $r \approx \sqrt{p}$, pick $s \approx \sqrt{p}$ so $rs < p$ and evaluate $A(k)$ at $s = \lfloor p/r \rfloor$ points to get

$$M_1 M_2 \cdots M_{p-1} \equiv_p A(1)A(r+1)A(2r+1)\cdots A((s-1)r+1)M_{sr+1}\cdots M_{p-1}.$$

Using standard product tree and multipoint evaluation techniques this takes $O(\mathsf{M}(p^{1/2})\log p) = p^{1/2}(\log p)^{2+o(1)}$ time.

Bostan-Gaudry-Schost: $p^{1/2}(\log p)^{1+o(1)}$ time.

The Hasse-Witt matrix of a hyperelliptic curve

Let $X_p/\mathbb{F}_p$ be a hyperelliptic curve $y^2 = f(x)$ of genus g (with $p \neq 2$).
As in the warmup, let f_k^n denote the coefficient of x^k in f^n.

The Hasse–Witt matrix of X_p is $W_p := [f_{pi-j}^n]_{ij} \in \mathbb{F}_p^{g\times g}$ with $n = (p-1)/2$.
By the same argument used in our warmup, have

$$a_p \equiv \sum_{i=1}^{g} f_{ip-i}^n = \operatorname{tr} W_p,$$

and for $p > 16g^2$ this uniquely determines $a_p \in \mathbb{Z}$.

In fact, as proved by Manin, we have

$$L_p(T) \equiv \det(I - TW_p) \bmod p.$$

One can define the Hasse-Witt matrix for curve over $\mathbb{F}_p$: it is the matrix of the p-power Frobenius acting on $H^1(X_p, \mathcal{O}_{X_p})$, and via Serre duality, the matrix of the Cartier-Manin operator acting on Ω_{X_p}.

Hyperelliptic average polynomial-time

As in our warmup, assume $f(0) \neq 0$ and define $v_k^n := [f_{k-d+1}^n, \ldots, f_k^n]$. The last g entries of v_{2n}^n form the first row of W_p, and we have

$$v_{2n}^n = -\left(\frac{f_0}{p}\right) V_0 M_1 \cdots M_{p-1} \bmod p \qquad \text{(where } V_0 = [0, \ldots, 0, 1]).$$

Compute the first row of W_p for good $p \leq B$ in $O(g^2 B (\log B)^{3+o(1)})$ time.

To get the remaining rows, consider the isomorphic curve $y^2 = f(x+a)$ whose Hasse-Witt matrix $W_p(a) = T(a) W_p T(-a)$ is conjugate to W_p via

$$T(a) := \left[\binom{j-1}{i-1} a^{j-1}\right]_{ij} \in \mathbb{F}_p^{g \times g}.$$

Given the first row of $W_p(a)$ for g distinct values of a we can compute all the rows of W_p. Total complexity is $O(g^3 B (\log B)^{3+o(1)})$, with an average complexity of $O(g^3 p^{4+o(1)})$, which is polynomial in both g and $\log p$.

Analytic rank data for elliptic/hyperelliptic curves

Genus 1 curves with $|\Delta_{\min}| \le 10^5: 17,247$ (exact count)

Genus 2 curves with $|\Delta_{\min}| \le 10^6: 66,158$ (lower bound)

Genus 3 curves with $|\Delta_{\min}| \le 10^7: 67,879$ (lower bound)

	genus 1		genus 2		genus 3	
rank	count	percent	count	percent	count	percent
0	6408	37,15	12131	18.34	7770	11.45
1	8586	49.78	30579	46.22	30840	45.47
2	2182	12.65	20561	31.08	25486	30.11
3	71	0.41	2877	4.35	3723	5.49
4	0	0.00	10	0.02	8	0.01

Genus 2 data includes all $y^2 + h(x)y = f(x)$ with $\|h\|_\infty \le 1, \|f\|_\infty \le 90$.
Genus 3 data includes all $y^2 + h(x)y = f(x)$ with $\|h\|_\infty \le 1, \|f\|_\infty \le 31$.

Genus 3 curves

The canonical embedding of a genus 3 curve into $\mathbb{P}^2$ is either

1. a degree-2 cover of a smooth conic (hyperelliptic case);
2. a smooth plane quartic (generic case).

Average polynomial-time implementations now available for all cases:

- rational hyperelliptic model $y^2 = f(x)$ [Harvey-S 2014].
- degree-2 cover of a smooth conic [Harvey-Massierer-S 2016].
- smooth plane quartic [Harvey-S 2017].

Essentially all prior work in genus 3 uses p-adic cohomology:

[Kedlaya 2001], [Gaudry-Gürel 2003], [Lauder 2004], [Kedlaya 2006], [Castryck-Denef-Vercauteren 2006], [Abbott-Kedlaya-Roe 2006], [Harvey 2007], [Hubrechts 2007], [Harvey 2010], [Hubrechts 2011], [Harrison 2012], [Tuitman-Pancrantz 2013], [Tuitman 2015], [Costa 2015], [Tuitman-Castryck 2016], [Shieh 2016]

The Hasse-Witt matrix of a smooth plane quartic

Let $X_p/\mathbb{F}_p$ be a smooth plane quartic defined by $f(x,y,z)=0$.
For $n \geq 0$ let $f^n_{i,j,k}$ denote the coefficient of $x^i y^j z^k$ in f^n.

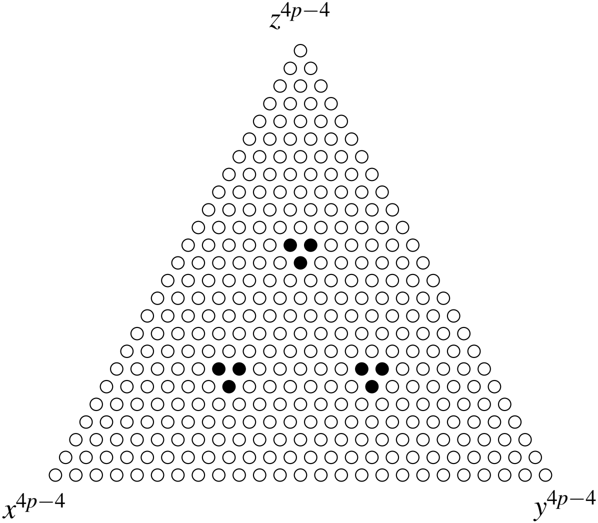
The Hasse–Witt matrix of X_p is the 3×3 matrix

$$W_p := \begin{bmatrix} f^{p-1}_{p-1,p-1,2p-2} & f^{p-1}_{2p-1,p-1,p-2} & f^{p-1}_{p-1,2p-1,p-2} \\ f^{p-1}_{p-2,p-1,2p-1} & f^{p-1}_{2p-2,p-1,p-1} & f^{p-1}_{p-2,2p-1,p-1} \\ f^{p-1}_{p-1,p-2,2p-1} & f^{p-1}_{2p-1,p-2,p-1} & f^{p-1}_{p-1,2p-2,p-1} \end{bmatrix}.$$

This case of smooth plane curves of degree $d > 4$ is similar.

More generally, given a singular plane model for any nice curve (equivalently, a defining polynomial for its function field) one can use the methods of Stohr-Voloch to explicitly determine W_p.

Target coefficients of f^{p-1} for $p = 7$:



Coefficient relations

Let $\partial_x = x\frac{\partial}{\partial x}$ (degree-preserving). The relations

$$f^{p-1} = f \cdot f^{p-2} \qquad \text{and} \qquad \partial_x f^{p-1} = -(\partial_x f) f^{p-2}$$

yield the relation

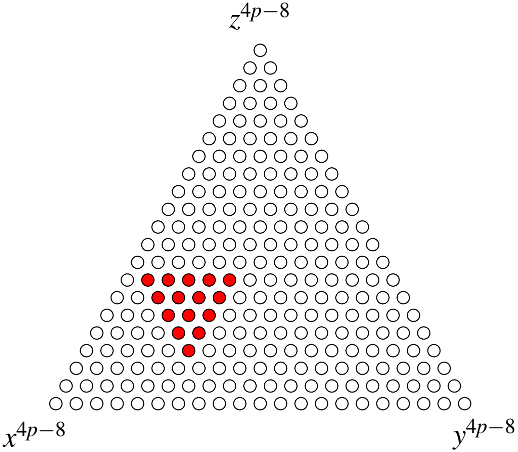
$$\sum_{i'+j'+k'=4} (i+i') f_{i',j',k'} f^{p-2}_{i-i',j-j',k-k'} = 0.$$

among nearby coefficients of f^{p-2} (a triangle of side length 5).

Replacing ∂_x by ∂_y yields a similar relation (replace $i+i'$ with $j+j'$).

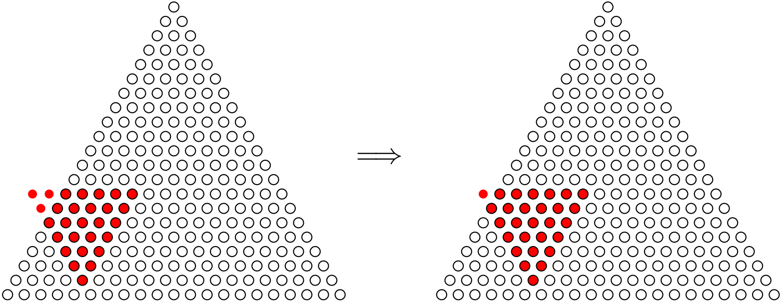
Coefficient triangle

For $p = 7$ with $i = 12, j = 5, k = 7$ the related coefficients of f^{p-2} are:



Moving the triangle

Now consider a bigger triangle with side length 7.
Our relations allow us to move the triangle around:



An initial “triangle” at the edge can be efficiently computed using coefficients of $f(x, 0, z)^{p-2}$.

Computing one Hasse-Witt matrix

Nondegeneracy: we need $f(1,0,0), f(0,1,0), f(0,0,1)$ nonzero and $f(0,y,z), f(x,0,z), f(x,y,0)$ squarefree (easily achieved for large p).

The basic strategy to compute W_p is as follows:

- There is a 28×28 matrix M_j that shifts our 7-triangle from y-coordinate j to $j+1$; its coefficients depend on j and f.
  In fact a 16×16 matrix M_i suffices (use smoothness of C).
- Applying the product $M_0 \cdots M_{p-2}$ to an initial triangle on the edge and applying a final adjustment to shift from f^{p-2} to f^{p-1} gets us one column of the Hasse-Witt matrix W_p.
- By applying the same product (or its inverse) to different initial triangles we can compute all three columns of W_p.

We have thus reduced the problem to computing $M_1 \cdots M_{p-2} \bmod p$, which we already know how to do, either in $p^{1/2}(\log p)^{1+o(1)}$ time, or in average polynomial time $(\log p)^{4+o(1)}$.

Cumulative timings for genus 3 curves

Time to compute $L_p(T) \bmod p$ for all good $p \leq B$.

B	spq-Costa-AKR	spq-HS	ghyp-MHS	hyp-HS	hyp-Harvey
2^{12}	18	1.4	0.3	0.1	1.3
2^{13}	49	2.4	0.7	0.2	2.6
2^{14}	142	4.6	1.7	0.5	5.4
2^{15}	475	9.4	4.6	1.0	12
2^{16}	1,670	21	11	2.1	29
2^{17}	5,880	47	27	5.3	74
2^{18}	22,300	112	62	14	192
2^{19}	78,100	241	153	37	532
2^{20}	297,000	551	370	97	1,480
2^{21}	1,130,000	1,240	891	244	4,170
2^{22}	4,280,000	2,980	2,190	617	12,200
2^{23}	16,800,000	6,330	5,110	1,500	36,800
2^{24}	66,800,000	14,200	11,750	3,520	113,000
2^{25}	244,000,000	31,900	28,200	8,220	395,000
2^{26}	972,000,000	83,300	62,700	19,700	1,060,000

(Intel Xeon E7-8867v3 3.3 GHz CPU seconds).