Genus 3 L-functions in average polynomial-time

Andrew V. Sutherland

Massachusetts Institute of Technology

March 28, 2017

Joint work with D. Harvey

L-functions and zeta functions

Given a smooth projective geometrically integral curve $X/\mathbb{Q}$ of genus g we wish to compute its L-function

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

where $L_p \in \mathbb{Z}[T]$ has degree at most $2g$. At primes p of good reduction the polynomial $L_p(T)$ is the numerator of the zeta function

$$Z(X_p/\mathbb{F}_p;T) := \exp\left(\sum_{k=1}^{\infty} \#X_p(\mathbb{F}_{p^k})T^k/k\right) = \frac{L_p(T)}{(1-T)(1-pT)}.$$

Ignoring bad primes, computing $L(X,s) \approx \sum_{n\leq N} a_n n^{-s}$ boils down to:

Given N, compute $L_p(T)$ for all good primes $p \leq N$.

In fact, for $p > \sqrt{N}$ we only need to know the trace of $L_p(T)$.

Algorithms to compute zeta functions

Given a curve $C/\mathbb{Q}$ of genus g, we want to compute the normalized L-polynomials $\overline{L}_p(T)$ at all good primes $p \le N$.

	complexity per prime (ignoring factors of $O(\log\log p)$)		
algorithm	$g=1$	$g=2$	$g=3$
point enumeration	$p\log p$	$p^2\log p$	$p^3(\log p)^2$
group computation	$p^{1/4}\log p$	$p^{3/4}\log p$	$p\log p$
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	$(\log p)^4$	$(\log p)^4$	$(\log p)^4$

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 available for the first case:

- rational hyperelliptic model [Harvey-S 2014];
- no rational hyperelliptic model [Harvey-Massierer-S 2016].

Here we will focus on the second case.

Prior work has all been based on p-adic cohomology:

[Lauder 2004], [Castryck-Denef-Vercauteren 2006], [Abott-Kedlaya-Roe 2006], [Harvey 2010], [Tuitman-Pancrantz 2013], [Tuitman 2015], [Costa 2015], [Tuitman-Castryck 2016], [Shieh 2016]

New algorithm

Let $C_p/\mathbb{F}_p$ 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 C_p is the 3×3 matrix

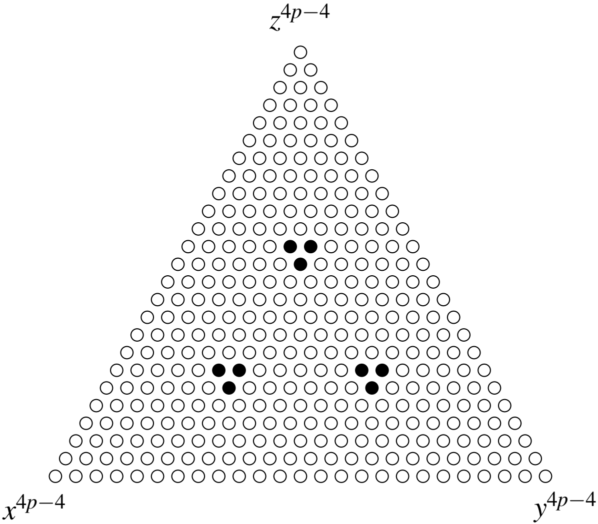
$$W_p := \begin{bmatrix} f^{p-1}_{p-1,p-1,2p-2} & f^{p-1}_{2p-1,p-1,p-2} & f^{p-1}_{p-1,2p-1,p-2} \\ f^{p-1}_{p-2,p-1,2p-1} & f^{p-1}_{2p-2,p-1,p-1} & f^{p-1}_{p-2,2p-1,p-1} \\ f^{p-1}_{p-1,p-2,2p-1} & f^{p-1}_{2p-1,p-2,p-1} & f^{p-1}_{p-1,2p-2,p-1} \end{bmatrix}.$$

This is the matrix of the p-power Frobenius acting on $H^1(C_p, \mathcal{O}_{C_p})$ (and the Cartier-Manin operator acting on the space of regular differentials). As proved by Manin, we have

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

Our strategy is to compute W_p then lift $L_p(T)$ from $(\mathbb{Z}/p\mathbb{Z})[T]$ to $\mathbb{Z}[T]$.

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

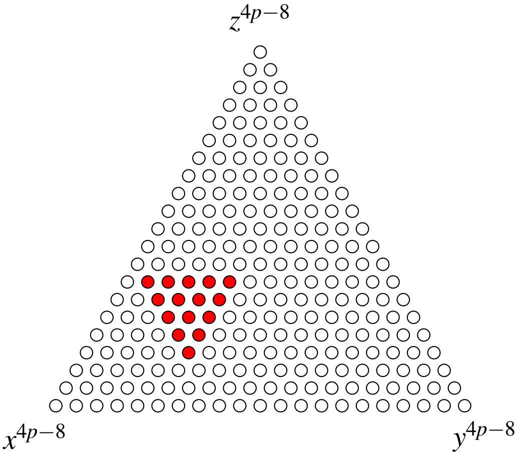
$$\sum_{\iota+\jmath+\kappa=4} (i+\iota) f_{\iota,b,\kappa} f^{p-2}_{i-\iota,j-\jmath,k-\kappa} = 0.$$

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

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

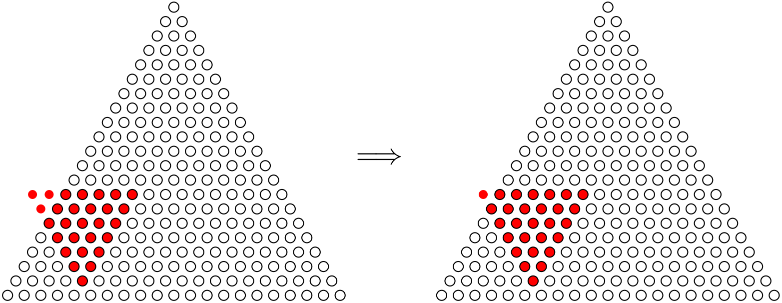
Coefficient triangle

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 $M_1 \cdots M_{p-2} \bmod p$.

An average polynomial-time algorithm

Now let $C/\mathbb{Q}$ be smooth plane quartic $f(x,y,z) = 0$ with $f \in \mathbb{Z}[x,y,z]$.
We want to compute W_p for all good $p \le N$.

Key idea

The matrices M_j do not depend on p; view them as integer matrices.
It suffices to compute $M_0 \cdots M_{p-2} \mod p$ for all good $p \le N$.

Using an *accumulating remainder tree* we can compute all of these partial products in time $O(N(\log N)^{3+o(1)})$.

This yields an average time of $O((\log p)^{4+o(1)})$ per prime to compute the W_p for all good $p \le N$.*

*We may need to skip $O(1)$ primes p where C_p is degenerate; these can be handled separately using an $\tilde{O}(p^{1/2})$ algorithm based on the same ideas.

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

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

and adjust the results as required.

Timings for genus 3 curves

	non-hyperelliptic		hyperelliptic	
N	`costa-AKR`	`avgpoly`	`harvey-K`	`avgpoly`
2^{12}	18.2	1.1	1.6	0.1
2^{13}	49.1	2.6	3.3	0.2
2^{14}	142	5.8	7.2	0.5
2^{15}	475	13.6	16.3	1.5
2^{16}	1,670	30.6	39.1	4.6
2^{17}	5,880	70.9	98.3	12.6
2^{18}	22,300	158	255	25.9
2^{19}	78,100	344	695	62.1
2^{20}	297,000	760	1,950	147
2^{21}	1,130,000	1,710	5,600	347
2^{22}	4,280,000	3,980	16,700	878
2^{23}	16,800,000	8,580	51,200	1,950
2^{24}	66,800,000	18,600	158,000	4,500
2^{25}	244,000,000	40,800	501,000	10,700
2^{26}	972,000,000	91,000	1,480,000	24,300

(Intel Xeon E7-8867v3 2.5 GHz CPU seconds).