Computing L-Series of genus 3 curves

Andrew V. Sutherland
Massachusetts Institute of Technology

July 1, 2017

Joint work with David Harvey; David Harvey and Maike Massierer; David Harvey; Andrew Booker, and David Platt.
The $L$-series of a curve

Let $X$ be a nice (smooth, projective, geometrically integral) curve of genus $g$ over $\mathbb{Q}$. The $L$-series of $X$ is the Dirichlet series

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

For primes $p$ of good reduction for $X$ we have the zeta function

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

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

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

where $\chi_p(T)$ is the charpoly of the Frobenius endomorphism of $\text{Jac}(X_p)$.
The Selberg class with polynomial Euler factors

The **Selberg class** $S_{\text{poly}}$ 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 $\varepsilon$, the completed $L$-function $\Lambda(s) := \gamma(s)L(s)$ satisfies the **functional equation**

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

where $Q > 0$, $\lambda_i > 0$, $\text{Re}(\mu_i) \geq 0$, $|\varepsilon| = 1$. Define $\deg L := 2 \sum_{i}^{r} \lambda_i$.

3. $a_1 = 1$ and $a_n = O(n^{\varepsilon})$ for all $\varepsilon > 0$ (**Ramanujan conjecture**).

4. $L(s) = \prod_{p} L_p(p^{-s})^{-1}$ for some $L_p \in \mathbb{Z}[T]$ with $\deg L_p \leq \deg L$ (has an **Euler product**).

The Dirichlet series $L_{\text{an}}(s, X) := L(X, s + \frac{1}{2})$ satisfies (3) and (4), and conjecturally lies in $S_{\text{poly}}$; 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 \( S^{\text{poly}} \) and \( a_p = b_p \) for all but finitely many primes \( p \), then \( A(s) = B(s) \).

**Corollary**

If \( L_{\text{an}}(s, X) \) lies in \( S^{\text{poly}} \) then it is completely determined by (any choice of) all but finitely many coefficients \( a_p \).

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

Let \( \Gamma_C(s) = 2(2\pi)^s \Gamma(s) \) and define \( \Lambda(X, s) := \Gamma_C(s)^8 L(X, s) \). Then

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

where the root number \( \varepsilon = \pm 1 \) and the analytic conductor \( N \in \mathbb{Z}_{\geq 1} \) are determined by the \( a_p \) values (we view these as definitions).
Testing the functional equation

Let $G(x)$ be the inverse Mellin transform of $\Gamma_C(s)^g = \int_0^\infty G(x)x^{s-1}dx$, and define

$$S(x) := \frac{1}{x} \sum a_n G(n/x),$$

so that $\Lambda(X, s) = \int_0^\infty S(x)x^{-s}dx$, and for all $x > 0$ we have

$$S(x) = \varepsilon S(N/x).$$

The function $G(x)$ decays rapidly, and for sufficiently large $c_0$ we have

$$S(x) \approx S_0(x) := \frac{1}{x} \sum_{n \leq c_0 x} a_n G(n/x),$$

with an explicit bound on the error $|S(x) - S_0(x)|$. 
Effective strong multiplicity one

Fix a finite set of small primes $S$ (e.g. $S = \{2\}$) 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 $p \in S$ (the coefficients of $L_p(T)$ are bounded).

Suppose we can compute $a_n$ for $n \leq \sqrt{M}$ whenever $p \nmid n$ for $p \in S$.

We now compute $\delta(x) := |S_0(x) - \varepsilon S_0(N/x)|$ with $x = c_1 \sqrt{N}$ for every possible choice of $\varepsilon$, $N$, and $L_p(T)$ for $p \in S$. If all but one choice makes $\delta(x)$ larger than our explicit error bound, we know the correct choice.

For a suitable choice of $c_1$ this is guaranteed to happen.\(^1\) 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.

---

\(^1\)Subject to our assumptions; if it does not happen then we have found an explicit counterexample to the conjectured Langlands correspondence.
Conductor bounds

The formula of Brumer and Kramer gives explicit bounds on the $p$-adic valuation of the algebraic conductor $N$ of $\text{Jac}(X)$:

$$v_p(N) \leq 2g + pd + (p - 1)\lambda_p(d),$$

where $d = \lfloor \frac{2g}{p-1} \rfloor$ and $\lambda_p(d) = \sum i d_i p^i$, with $d = \sum d_i p^i$ with $0 \leq d_i < p$.

<table>
<thead>
<tr>
<th>$g$</th>
<th>$p = 2$</th>
<th>$p = 3$</th>
<th>$p = 5$</th>
<th>$p = 7$</th>
<th>$p &gt; 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>10</td>
<td>9</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>21</td>
<td>11</td>
<td>13</td>
<td>6</td>
</tr>
</tbody>
</table>

For $g \leq 2$ these bounds are tight (see www.lmfdb.org for examples).

For hyperelliptic curves $N$ divides $\Delta(X)$. Smooth plane curves?
Algorithms to compute zeta functions

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

<table>
<thead>
<tr>
<th>algorithm</th>
<th>( g = 1 )</th>
<th>( g = 2 )</th>
<th>( g = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>point enumeration</td>
<td>( p \log p )</td>
<td>( p^2 \log p )</td>
<td>( p^3 (\log p)^2 )</td>
</tr>
<tr>
<td>group computation</td>
<td>( p^{1/4} \log p )</td>
<td>( p^{3/4} \log p )</td>
<td>( p (\log p)^2 )</td>
</tr>
<tr>
<td>( p )-adic cohomology</td>
<td>( p^{1/2} (\log p)^2 )</td>
<td>( p^{1/2} (\log p)^2 )</td>
<td>( p^{1/2} (\log p)^2 )</td>
</tr>
<tr>
<td>CRT (Schoof-Pila)</td>
<td>( (\log p)^5 )</td>
<td>( (\log p)^8 )</td>
<td>( (\log p)^{12?} )</td>
</tr>
<tr>
<td>average poly-time</td>
<td>( (\log p)^4 )</td>
<td>( (\log p)^4 )</td>
<td>( (\log p)^4 )</td>
</tr>
</tbody>
</table>

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

The bottom line: it all comes down to computing \( a_p \)'s.
Let $X : y^2 = f(x)$ with $\deg f = 3, 4$ and $f(0) \neq 0$, and let $f^n_k$ be the coefficient of $x^k$ in $f^n$. Then $a_p \equiv f_{p-1}^{(p-1)/2} \mod p$ for all good $p$.

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

$$kf_0 f^n_k = \sum_{1 \leq i \leq d} (n + 1) - k)f_i f^{n-i}_k,$$

for all $k, n \geq 0$. Suppose for simplicity $\deg f = 3$, and define

$$v^n_k := [f^n_{k-2}, f^n_{k-1}, f^n_k], \quad M^n_k := \begin{bmatrix} 0 & 0 & (3n + 3 - k)f_3 \\ kf_0 & 0 & (2n + 2 - k)f_2 \\ 0 & kf_0 & (n + 1 - k)f_1 \end{bmatrix},$$

so that we have the recurrence $v^n_k = \frac{1}{kf_0} v^n_{k-1} M^n_k$. 
Warmup: average polynomial-time in genus 1

We then have

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

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

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

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

is an integer matrix whose entries do not depend on $p = 2n + 1$, and

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

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

\[
M_0 \mod m_1 \\
M_0M_1 \mod m_2 \\
M_0M_1M_2 \mod m_3 \\
M_0M_1M_2M_3 \mod m_4 \\
\vdots \\
M_0M_1 \cdots M_{n-2}M_{n-1} \mod m_n
\]

multiply adjacent pairs and recursively compute

\[
(M_0M_1) \mod m_2m_3 \\
(M_0M_1)(M_2M_3) \mod m_4m_5 \\
\vdots \\
(M_0M_1) \cdots (M_{n-2}M_{n-1}) \mod m_n
\]

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 $3 \times 3$ 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 in $O(M(B \log B)) = B(\log B)^2 + o(1)$.

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 $a_p$ in genus 1, but it turns out to be much faster than any other method currently available in genus 3.
Efficiently handling a single prime

Simply computing $V_0M_1 \cdots M_{p-1}$ modulo $p$ is surprisingly quick (faster than semi-naïve point-counting); it takes $p(\log p)^{1+o(1)}$ time. But we can do better.

Viewing $M_k \mod p$ 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 $r \approx \sqrt{p}$ and then instantiate $A(k)$ at roughly $r$ points to get

$$M_1M_2 \cdots M_{p-1} \equiv_p A(1)A(r+1)A(2r+1) \cdots A(p-r).$$

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

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

New result (joint with Harvey): smooth plane quartics.

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]

Current implementations of these algorithms are all $O(p^{1+o(1)})$. 
The Hasse-Witt matrix of a hyperelliptic curve

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

The Hasse–Witt matrix of $X_p$ is $W_p := [f^n_{p-i-j}]_{ij} \in \mathbb{F}_p^{g \times g}$ with $n = (p - 1)/2$. In genus $g = 3$ we have

$$W_p := \begin{bmatrix} f^n_{p-1} & f^n_{p-2} & f^n_{p-3} \\ f^n_{2p-1} & f^n_{2p-2} & f^n_{2p-3} \\ f^n_{3p-1} & f^n_{3p-2} & f^n_{3p-3} \end{bmatrix}.$$

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 regular differentials). As proved by Manin, we have

$$L_p(T) \equiv \det(I - TW_p) \mod p;$$

in particular, $a_p \equiv \text{tr} W_p \mod p$. For $p > 144$ this yields $a_p \in [-6\sqrt{p}, 6\sqrt{p}]$. 

Hyperelliptic average polynomial-time

As in our warmup, assume \( f(0) \neq 0 \) and define \( v^n_k := [f^n_{k-d+1}, \ldots, f^n_k] \).

The last \( g \) entries of \( v^n_{2n} \) form the first row of \( W_p \), and we have

\[
v^n_{2n} = - \left( \frac{f_0}{p} \right) V_0 M_1 \cdots M_{p-1} \mod p \quad \text{(where } V_0 = [0, \ldots, 0, 1]).\]

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 \( W_p(a) = T(a)W_pT(-a) \) is conjugate to \( W_p \) via

\[
T(a) := \begin{bmatrix} (j - 1) a^{i-1} \end{bmatrix}_{ij} \in \mathbb{F}_p^{g \times g}.
\]

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 \( O(g^3 B (\log B)^{3+o(1)}) \).
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 \( f_{i,j,k}^n \) denote the coefficient of \( x^i y^j z^k \) in \( f^n \).

The Hasse–Witt matrix of \( X_p \) is the \( 3 \times 3 \) matrix

\[
W_p := \begin{bmatrix}
  f_{p-1,p-1,2p-2}^p & f_{2p-1,p-1,p-2}^p & f_{p-1,2p-1,p-2}^p \\
  f_{p-1}^p & f_{2p-1,p-1,p-1}^p & f_{p-2,2p-1,p-1}^p \\
  f_{p-1}^p & f_{2p-1,p-2,p-1}^p & f_{p-1,2p-2,p-1}^p
\end{bmatrix}.
\]

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

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

yield the relation

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

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

Replacing $\partial_x$ by $\partial_y$ yields a similar relation (replace $i + i'$ with $j + j'$).
For $p = 7$ with $i = 12, j = 5, k = 7$ the related coefficients of $f^{p-2}$ are:

$$x^{4p-8}, \quad y^{4p-8}, \quad z^{4p-8}$$
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 \(28 \times 28\) 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 \(16 \times 16\) 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 \(M_1 \cdots M_{p-2} \mod p\), 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 $L_p(T) \mod p$ for all good $p \leq B$.

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

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