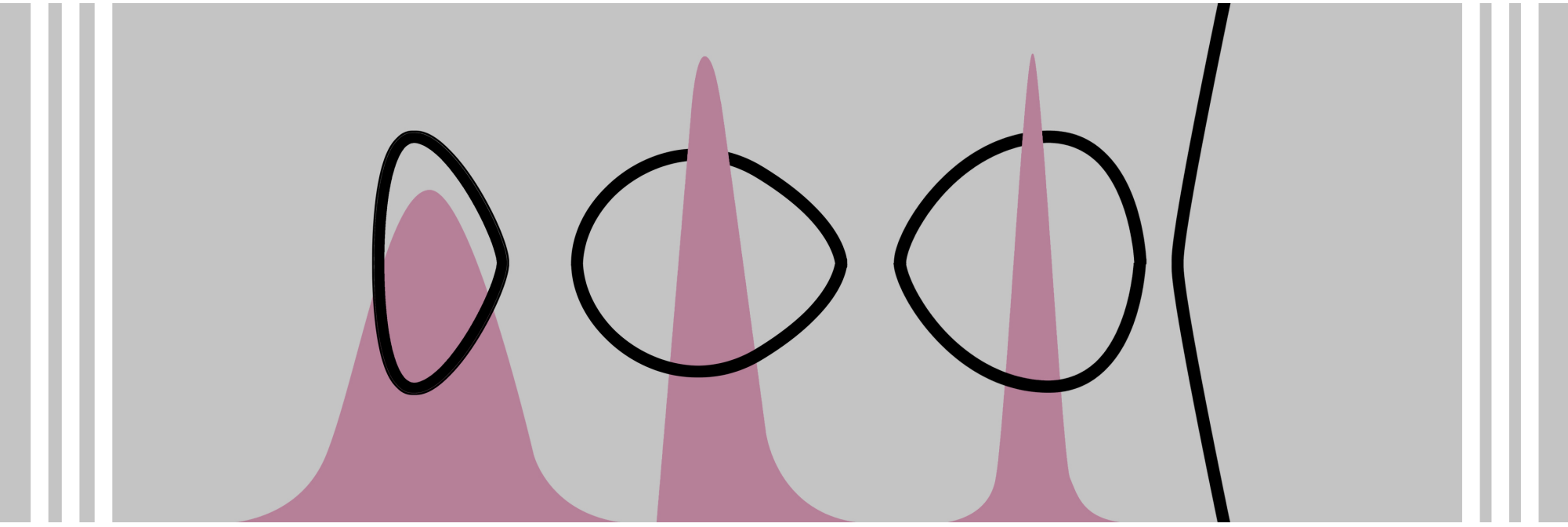
Computing L-series of geometrically hyperelliptic curves of genus three



David Harvey, Maike Massierer, Andrew V. Sutherland

The zeta function

Let

- $C/\mathbb{Q}$ be a smooth projective curve of genus 3
- p be a prime of good reduction
- C_p be the reduction of C modulo p

$$Z_p(T) = \exp\left(\sum_{k=1}^{\infty} \frac{\#C_p(\mathbb{F}_{p^k})}{k} T^k\right) = \frac{L_p(T)}{(1-T)(1-pT)}$$

$$L_p(T) = 1 + a_1 T + a_2 T^2 + a_3 T^3 + p a_2 T^4 + p^2 a_1 T^5 + p^3 T^6 \in \mathbb{Z}[T]$$

The zeta function

Let

- $C/\mathbb{Q}$ be a smooth projective curve of genus 3
- p be a prime of good reduction
- C_p be the reduction of C modulo p

$$Z_p(T) = \exp\left(\sum_{k=1}^{\infty} \frac{\#C_p(\mathbb{F}_{p^k})}{k} T^k\right) = \frac{L_p(T)}{(1-T)(1-pT)}$$

$$L_p(T) = 1 + a_1 T + a_2 T^2 + a_3 T^3 + p a_2 T^4 + p^2 a_1 T^5 + p^3 T^6 \in \mathbb{Z}[T]$$

Goal: Given a curve C and a bound N, compute $L_p(T)$ for all $p < N$ of good reduction.

Motivation

- Collecting data for the Sato–Tate conjecture
- Computing the first N terms of the L-series

$$L(C,s) = \prod_p L_p(p^{-s})^{-1} = \sum_{n \geq 1} c_n n^{-s}$$

Motivation

- Collecting data for the Sato–Tate conjecture
  [it's okay to leave out a few primes]
- Computing the first N terms of the L-series

$$L(C,s) = \prod_p L_p(p^{-s})^{-1} = \sum_{n \geq 1} c_n n^{-s}$$

[also need $L_p(T)$ at primes of bad reduction]

Curves of genus 3 over $\mathbb{Q}$

1. Geometrically hyperelliptic [double cover of a conic]
   $g(x,y) = 0,\ w^2 = f(x,y)$
   $f, g \in \mathbb{Z}[x,y],\ \deg(g) = 2,\ \deg(f) = 4$

2. Smooth plane quartic

Curves of genus 3 over $\mathbb{Q}$

1. Geometrically hyperelliptic [double cover of a conic]
   $g(x,y) = 0,\ w^2 = f(x,y)$
   $f, g \in \mathbb{Z}[x,y],\ \deg(g) = 2,\ \deg(f) = 4$

   If the conic has a rational point: ordinary hyperelliptic
   $y^2 = h(x)$
   $h \in \mathbb{Z}[x],\ \deg(h) = 7, 8$

2. Smooth plane quartic

Curves of genus 3 over $\mathbb{Q}$

1. Geometrically hyperelliptic [double cover of a conic]
   $g(x, y) = 0,\ w^2 = f(x, y)$
   $f, g \in \mathbb{Z}[x, y],\ \deg(g) = 2,\ \deg(f) = 4$
   [this work]

   If the conic has a rational point: ordinary hyperelliptic
   $y^2 = h(x)$
   $h \in \mathbb{Z}[x],\ \deg(h) = 7, 8$
   [Harvey, Sutherland (2014, 2016)]
2. Smooth plane quartic
   [Harvey, Sutherland (in progress)]

Example

- Ordinary hyperelliptic curve:

$$y^2 = 2x^8 - 2x^7 + 3x^6 - 2x^5 - 4x^4 + 2x^3 + 2x + 2$$

- Double cover of a pointless conic:

$$\begin{aligned} 0 &= x^2 + y^2 + 1 \\ w^2 &= x^4 - 2x^2y^2 - 2y^4 - x^3 - 2x^2y - xy^2 \\ &\quad - y^3 - x^2 - xy - y^2 + x + 1 \end{aligned}$$

Theorem

There exists an explicit deterministic algorithm with
Input: f, g, N
Output: $L_p(T)$ for good $p < N$
Running time: $N \log^{3+o(1)} N$ bit operations
[ignoring dependence on size of coefficients of f, g]

Average time per prime: $\log^{4+o(1)} N$

Theorem

There exists an explicit deterministic algorithm with
Input: f, g, N
Output: $L_p(T)$ for good $p < N$
Running time: $N \log^{3+o(1)} N$ bit operations
[ignoring dependence on size of coefficients of f, g]

Average time per prime: $\log^{4+o(1)} N$

Theoretical result: Harvey (2015)
Practical algorithm: **this work**
[not quite polytime, but the polytime part dominates]

Theorem

There exists an explicit deterministic algorithm with
Input: f, g, N
Output: $L_p(T)$ for good $p < N$
Running time: $N \log^{3+o(1)} N$ bit operations
[ignoring dependence on size of coefficients of f, g]

Average time per prime: $\log^{4+o(1)} N$

Theoretical result: Harvey (2015)
Practical algorithm: **this work**

[not quite polytime, but the polytime part dominates]

[this approach is **global**: we treat all p simultaneously]

[one p at a time: get ordinary hyperelliptic model]

Overview of the algorithm

1. Compute model $C' : y^2 = h(x)$
   over quadratic extension K of $\mathbb{Q}$
2. Compute Hasse–Witt matrices of $C'_{\mathfrak{p}}$
   where $\mathfrak{p} \mid p$, for all $p < N$ simultaneously
   [accumulating remainder tree algorithm]
3. Compute $L_p(T) \mod p$ (split prime)
   or $L_p(T)L_p(-T) \mod p$ (inert prime)
   for each $p < N$
   [reversed characteristic polynomial]
4. Lift to $L_p(T) \in \mathbb{Z}[T]$ for each $p < N$
   [baby step giant step in Jacobian]

Hyperelliptic model

Given: $C : w^2 = f(x, y),\ g(x, y) = 0$ over $\mathbb{Z}$

$[\deg 4]$ $[\deg 2]$

Denote by Q the conic $g = 0$

Compute: Quadratic field K and hyperelliptic model for $C' = C \times_{\mathbb{Q}} K$ [base change of C to K] :

- Let $K = \mathbb{Q}(\sqrt{D})$ s.t. $\exists P \in Q(K)$
- Parametrize Q w.r.t. P
- Plug parametrization into $w^2 = f(x, y)$
- Obtain $C' : y^2 = h(x), h \in O_K[x], \deg(h) = 8$
  [after multiplying by denominators, possibly applying birational transformation; enforce technical conditions]

Example

- Double cover of a pointless conic:

$$\begin{aligned} 0 &= x^2 + y^2 + 1 \\ w^2 &= x^4 - 2x^2y^2 - 2y^4 - x^3 - 2x^2y - xy^2 \\ &\quad - y^3 - x^2 - xy - y^2 + x + 1 \end{aligned}$$

- Parametrization over $K = \mathbb{Q}(i)$ $[D = -1]$:

$$\begin{aligned} y^2 &= (3-2i)x^8 + (2-4i)x^7 + (-4-4i)x^6 \\ &\quad + (2-4i)x^5 + 2x^4 + (-2-4i)x^3 \\ &\quad + (-4+4i)x^2 + (-2-4i)x + (3+2i) \end{aligned}$$

Hasse–Witt matrices

Given: $C' : y^2 = h(x)$ over O_K, $h(x) = \sum_{i=0}^{8} h_i x^i$

For odd unramified $p < N$, let $\mathfrak{p} \,|\, p$ prime ideal of K.

Compute: Hasse–Witt matrix of C' at $\mathfrak{p}$:

$$W_{\mathfrak{p}} \in (O_K/\mathfrak{p})^{3\times 3}, \quad (W_{\mathfrak{p}})_{i,j} = h_{pi-j}^{(p-1)/2} \bmod \mathfrak{p}$$

Hasse–Witt matrices

Given: $C' : y^2 = h(x)$ over O_K, $h(x) = \sum_{i=0}^{8} h_i x^i$

For odd unramified $p < N$, let $\mathfrak{p} \,|\, p$ prime ideal of K.

Compute: Hasse–Witt matrix of C' at $\mathfrak{p}$:

$$W_{\mathfrak{p}} \in (O_K/\mathfrak{p})^{3\times 3}, \quad (W_{\mathfrak{p}})_{i,j} = h_{pi-j}^{(p-1)/2} \bmod \mathfrak{p}$$

Proposition: It is enough to compute the first rows of the three Hasse–Witt matrices associated to three translates $y^2 = h(x+\beta)$ of the curve.

[solve a 9×9 linear system]

Recurrence for first row

Compute: $\left(h_{p-1}^{(p-1)/2}, h_{p-2}^{(p-1)/2}, h_{p-3}^{(p-1)/2}\right) \bmod p$

[reduce mod $\mathfrak{p}$ to get first row of Hasse–Witt matrix]

Let $v_k = \left(h_{k-7}^{(p-1)/2}, \ldots, h_k^{(p-1)/2}\right) \in (O_K)^8$

Derive recurrence $v_k = \frac{1}{2kh_0} v_{k-1} M_k \bmod p$ for $M_k \in (O_K)^{8\times 8}$

[Bostan, Gaudry, Schost (2007)]

Recurrence for first row

Compute: $\left(h_{p-1}^{(p-1)/2}, h_{p-2}^{(p-1)/2}, h_{p-3}^{(p-1)/2}\right) \bmod p$

[reduce mod $\mathfrak{p}$ to get first row of Hasse–Witt matrix]

Let $v_k = \left(h_{k-7}^{(p-1)/2}, \ldots, h_k^{(p-1)/2}\right) \in (O_K)^8$

Derive recurrence $v_k = \frac{1}{2kh_0} v_{k-1} M_k \bmod p$ for $M_k \in (O_K)^{8\times 8}$

[Bostan, Gaudry, Schost (2007)]

Proposition: The first row can be easily computed from

$$v_0 M_1 \cdots M_{p-1} \bmod p$$

Recurrence for first row

Compute: $\left(h_{p-1}^{(p-1)/2}, h_{p-2}^{(p-1)/2}, h_{p-3}^{(p-1)/2}\right) \bmod p$

[reduce mod $\mathfrak{p}$ to get first row of Hasse–Witt matrix]

Let $v_k = \left(h_{k-7}^{(p-1)/2}, \ldots, h_k^{(p-1)/2}\right) \in (O_K)^8$

Derive recurrence $v_k = \frac{1}{2kh_0} v_{k-1} M_k \bmod p$ for $M_k \in (O_K)^{8\times 8}$

[Bostan, Gaudry, Schost (2007)]

Proposition: The first row can be easily computed from

$$v_0 M_1 \cdots M_{p-1} \bmod p$$

Evaluate product using accumulating remainder tree algorithm [Costa, Gerbicz, Harvey (2014)]

L-polynomials modulo p

Given: Hasse–Witt matrix $W_{\mathfrak{p}}$ for $\mathfrak{p} \mid p,\ p < N$

Let $L'_{\mathfrak{p}}$ be the L-polynomial of C' at $\mathfrak{p}$.

Compute:

p split: $C'_{\mathfrak{p}} \cong C_p$ over $O_K/\mathfrak{p} \cong \mathbb{F}_p$

$$L'_{\mathfrak{p}}(T) = \det(I - TW_{\mathfrak{p}}) \bmod \mathfrak{p}$$

determines $L_p(T) \bmod p$

p inert: $O_K/\mathfrak{p} \cong \mathbb{F}_{p^2}$

$$L'_{\mathfrak{p}}(T) = \det(I - TW_{\mathfrak{p}}W_{\mathfrak{p}}^{(p)}) \bmod \mathfrak{p}$$

determines $L_p(T)L_p(-T) = L'_{\mathfrak{p}}(T^2) \bmod p$

[we lost information when passing from C to C':
we computed the zeta function of C_p over $\mathbb{F}_{p^2}$]

Lifting

Given: $L_p(T) \bmod p$ or $L_p(T)L_p(-T) \bmod p$

Compute: $L_p(T) \in \mathbb{Z}[T]$

[recall: $L_p(T) = 1 + a_1T + a_2T^2 + a_3T^3 + pa_2T^4 + p^2a_1T^5 + p^3T^6$]

- Compute model $C_p : y^2 = h(x)$ over $\mathbb{F}_p$
- Use Weil bounds on coefficients of $L_p(T)$:
  $|a_1| \leq 6p^{1/2}, \ |a_2| \leq 15p, \ |a_3| \leq 20p^{3/2}$
- Use $\#\operatorname{Jac}(C_p)(\mathbb{F}_p) = L_p(1)$
  $\#\operatorname{Jac}(\tilde{C}_p)(\mathbb{F}_p) = L_p(-1)$ [quadratic twist]

Lifting

Given: $L_p(T) \bmod p$ or $L_p(T)L_p(-T) \bmod p$

Compute: $L_p(T) \in \mathbb{Z}[T]$

[recall: $L_p(T) = 1 + a_1T + a_2T^2 + a_3T^3 + pa_2T^4 + p^2a_1T^5 + p^3T^6$]

- Compute model $C_p : y^2 = h(x)$ over $\mathbb{F}_p$
- Use Weil bounds on coefficients of $L_p(T)$:
  $|a_1| \leq 6p^{1/2}, \ |a_2| \leq 15p, \ |a_3| \leq 20p^{3/2}$
- Use $\#\operatorname{Jac}(C_p)(\mathbb{F}_p) = L_p(1)$
  $\#\operatorname{Jac}(\tilde{C}_p)(\mathbb{F}_p) = L_p(-1)$ [quadratic twist]
- Las Vegas algorithm to compute the order of $\operatorname{Jac}(C_p)(\mathbb{F}_p)$ and $\operatorname{Jac}(\tilde{C}_p)(\mathbb{F}_p)$
  [compute orders of elements via baby step giant step]

Lifting

Given: $L_p(T) \bmod p$ or $L_p(T)L_p(-T) \bmod p$

Compute: $L_p(T) \in \mathbb{Z}[T]$

[recall: $L_p(T) = 1 + a_1T + a_2T^2 + a_3T^3 + pa_2T^4 + p^2a_1T^5 + p^3T^6$]

- Compute model $C_p : y^2 = h(x)$ over $\mathbb{F}_p$
- Use Weil bounds on coefficients of $L_p(T)$:
  $|a_1| \leq 6p^{1/2}, \ |a_2| \leq 15p, \ |a_3| \leq 20p^{3/2}$
- Use $\#\operatorname{Jac}(C_p)(\mathbb{F}_p) = L_p(1)$
  $\#\operatorname{Jac}(\tilde{C}_p)(\mathbb{F}_p) = L_p(-1)$ [quadratic twist]
- Las Vegas algorithm to compute the order of $\operatorname{Jac}(C_p)(\mathbb{F}_p)$ and $\operatorname{Jac}(\tilde{C}_p)(\mathbb{F}_p)$
  [compute orders of elements via baby step giant step]
- Time $p^{1/4+o(1)}$ [but negligible in practice]

Implementation

- Practical considerations
  - [Running time is dominated by remainder trees]
    Specialized implementation of FFT-based multiplication for matrix-matrix and matrix-vector multiplications over $O_K = \mathbb{Z}[\alpha]$
  - [Extremely memory intensive]
    Remainder tree $\longrightarrow$ remainder forest
    [saves $\log N$ factor space, constant factor time]

Implementation

- Practical considerations
  - [Running time is dominated by remainder trees]
    Specialized implementation of FFT-based multiplication for matrix-matrix and matrix-vector multiplications over $O_K = \mathbb{Z}[\alpha]$
  - [Extremely memory intensive]
    Remainder tree $\longrightarrow$ remainder forest
    [saves $\log N$ factor space, constant factor time]
- Implementation setting
  - Based on ordinary hyperelliptic case
  - In C, using GMP and FFT library
  - Single core 2.5 GHz [tricky to parallelize]
  - Server with 1088 GB RAM

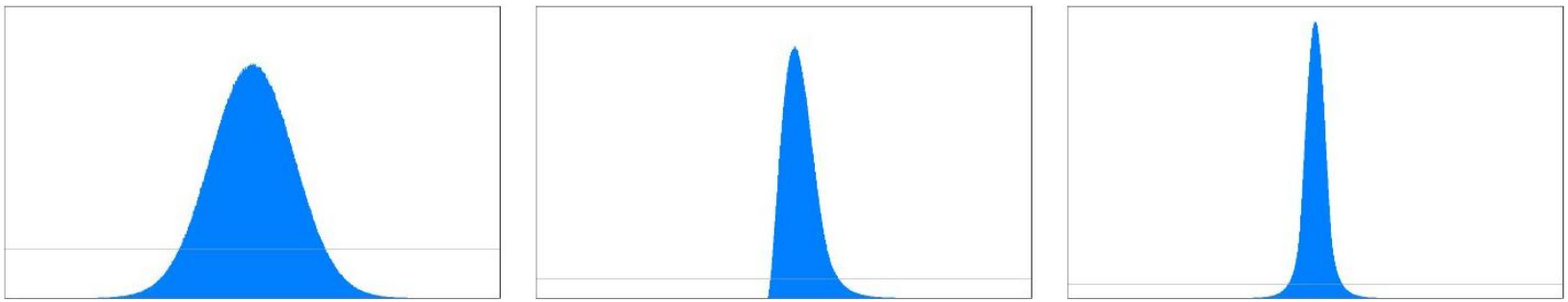
Implementation results for $N = 2^{30}$

		ordinary hyperelliptic	cover of a pointless conic
new algorithm	time	8 days	**23 days**
	space	288 GB	**480 GB**
	lift	13 hours	**20 hours**
`hypellfrob`	time	3 years	

`hypellfrob:` one prime at a time with previously best algorithm [Harvey (2007)]

Sato–Tate histograms

We get the generic distribution for $\mathrm{USp}(6)$:

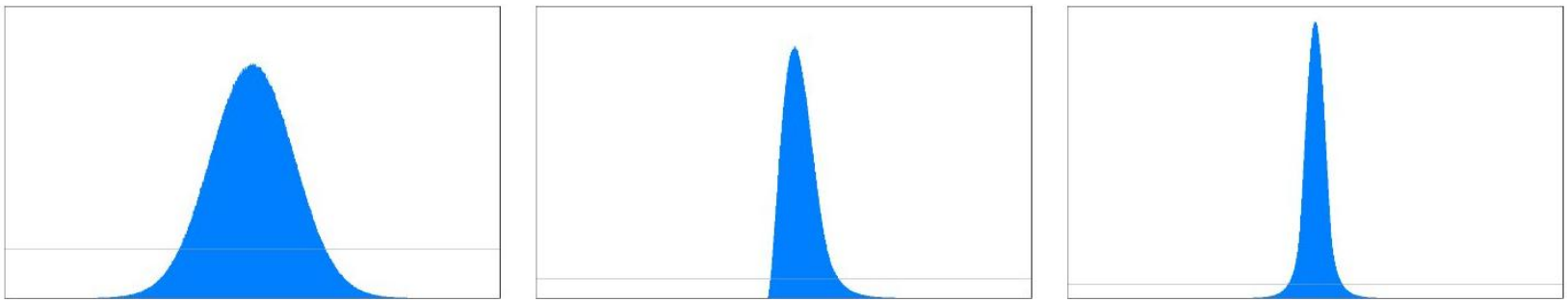


[http://math.mit.edu/~drew/ants12histograms.html]

This is as expected, but we are curious to see which distributions we might obtain for other curves.

Sato–Tate histograms

We get the generic distribution for $\mathrm{USp}(6)$:



[http://math.mit.edu/~drew/ants12histograms.html]

This is as expected, but we are curious to see which distributions we might obtain for other curves.

Thank you!