Numerical verification of BSD

for hyperelliptics of genus 2 & 3, and beyond...

Raymond van Bommel (Universiteit Leiden)

PhD project under supervision of:
David Holmes (Leiden)
Fabien Pazuki (Bordeaux/Copenhagen)

7 September 2017

Outline

In Magma the speaker implemented an algorithm to numerically verify BSD for the Jacobian J of an hyperelliptic curve $C/\mathbb{Q}$ of higher genus, i.e. the algorithm calculates (up to squares)

- $\lim_{s\to 1}(s-1)^{-r}L(J,s)$,
- the real period P_J,
- the regulator R_J,
- the Tamagawa numbers c_p, and
- the size of $J(\mathbb{Q})_{\mathrm{tors}}$,

then it uses the BSD formula

$$\lim_{s\to 1}(s-1)^{-r}L(J,s) = \frac{P_J R_J \cdot |Ш(J)| \cdot \prod_p c_p}{|J(\mathbb{Q})_{\mathrm{tors}}|^2}$$

to predict the size of $Ш(J)$ (up to squares).

List of results

The algorithm numerically verified BSD for:

- all elliptic curves $y^2 = x^3 + ax + b$ with $a, b \in \{-15, \ldots, 15\}$, comparing it with existing routines in `Magma`;
- most hyperelliptic curves of genus 2 with low conductor from the 'Empirical evidence' paper (Flynn et al., 2001), comparing it with the results from this paper;
- all 300 hyperelliptics $C : y^2 = x^5 + ax^4 + bx^3 + cx^2 + dx + e$ with $a, b, c, d, e \in \{-10, \ldots, 10\}$ and $\Delta(C) \leq 10^5$, except for 30 examples;
- 29 hyperelliptics curves of genus 3 (verification up to squares)
  $$C : y^2 = x^7 + ax^6 + bx^5 + cx^4 + dx^3 + ex^2 + fx + g$$
  with $a, b, c, d, e, f, g \in \{-3, \ldots, 3\}$ and $\Delta(C) \leq 10^7$.

In all cases, except for the ones already considered by Flynn et al., the predicted order of $Ш(J)$ is 1.

List of exceptions

The algorithm failed for

- several examples for which no regular model could be computed by `Magma` at some prime p of bad reduction; (might be resolved partially using the new method)
- for genus 3: some examples for which the conductor was big, which prolongs the calculation of the L-function and period;
- the curve $x^5 - 4x^4 + 8x^3 - 8x^2 + 4x - 1$ for which the height code takes too long to excute;
- the curve $x^5 - 3x^4 + 6x^3 - 6x^2 + 4^x - 1$ for which the L-function code takes too long to execute.

Universiteit Leiden

Runtimes

For the following curves

- H_1: genus 2, rank 0, discriminant 5^{16} (from 'Empirical' paper)
- H_2: genus 2, rank 1, discriminant 62720
- H_3: genus 3, rank 1, discriminant -1523712

we recorded the following runtimes (in seconds):

	H_1	H_2	H_3
$\lim_{s\to 1}(s-1)^{-r}L(J,s)$	8.930	7.520	173.5
period P_J	36.33	34.34	64.46
regulator R_J	0.930	142.6	294.23
Tamagawa numbers c_p	0.040	0.040	0.070
$\lvert J(\mathbb{Q})_{\mathrm{tors}}\rvert$	0.130	0.010	N/A

How to calculate $\lim_{s\to 1}(s-1)^{-r}L(J,s)$?

For the algebraic rank:

- upper bounds: 2-Selmer groups
- lower bounds: point searching

For the L-function and conductor (due to Tim Dokchitser):

- most places: point counting to get local factor
- other places: guess using the functional equation

Problem: the runtime seems to increase quickly as the conductor increases.

Possible solution: use the methods of Sutherland.

How to calculate the regulator R_J?

For the regulator:

- find generators for the free part of $J(\mathbb{Q})$
- calculate height pairing (due to Holmes and Müller)

Problem: the bound for the naive height is big. In practice, this might give rise to an error factor, which is a rational square.

Problem: the higher the genus gets, the harder it is enumerate all points of bounded height.

How to calculate the Tamagawa numbers c_p?

As seen in Morgan's course: calculate an explicit regular model and do the computations there.

The speaker's main contribution here is a `Magma` package that computes the Galois action on the geometric component group (which is already included in the `RegularModel` package), and uses this to compute the Tamagawa numbers.

This has been used to calculate Tamagawa numbers for all but 54 genus 2 curves in the LMFDB.

How to calculate $|J(\mathbb{Q})_{\mathrm{tors}}|$?

For the torsion:

- lower bounds: finding points
- upper bounds: counting points on reduction mod p

If the lower and upper bounds do not match, the error induced will be at worst a rational square.

How to calculate the real period P_J? (1/5)

For a standard basis $\frac{dx}{y}, \frac{xdx}{y}, \ldots, \frac{x^{g-1}dx}{y}$ of the differentials, and for a symplectic basis $\gamma_1, \ldots, \gamma_{2g}$ of $H^1(J(\mathbb{C}), \mathbb{Z})$ calculated by `Magma`, there is a `Magma` routine `BigPeriodMatrix` due to Van Wamelen that calculates the matrix

$$M = \left(\int_{\gamma_i} \frac{x^{j-1}dx}{y} \right)_{i=1,\ldots,2g,\, j=1,\ldots g}.$$

The columns of $M + \overline{M}$ span a lattice inside $\mathbb{R}^g$. The covolume of this lattice is the real period, up to a certain *correction factor*.

The differential $\frac{dx}{y} \wedge \ldots \wedge \frac{x^{g-1}dx}{y}$ is not a Néron differential in general. To correct for this, we need to find how far it is away from being a Néron differential.

Universiteit Leiden

How to calculate the real period P_J? (2/5)

For odd primes of good reduction, it is alright, but for the other primes p we do the following calculation (cf. Flynn et al.):

1. we calculate a regular model $\mathcal{C}/\mathbb{Z}_{(p)}$;
2. for each $i = 0, \ldots, g-1$ and each irreducible component E of the special fibre $\mathcal{C}_{\mathbb{F}_p}$, we check if $\frac{x^i dx}{y}$ has a pole on E and multiply by p if necessary;
3. for each linear combination $D = \sum_{i=0}^{g-1} c_i \frac{x^i dx}{y}$, with $c_i \in \{0, \ldots, p-1\}$ not all zero, and each component E of $\mathcal{C}_{\mathbb{F}_p}$, we check if D vanishes on E. We adjust the basis, in case one such D vanishes on the whole special fibre.

How to calculate the real period P_J? (3/5)

Question: given a differential, regular on $C/\mathbb{Q}$, how to calculate its order of vanishing on components of the special fibre $\mathcal{C}_{\mathbb{F}_p}$?

Answer: Classically, for smooth $C/\mathbb{Q}$, there is an isomorphism

$$\Omega^1_{J/\mathbb{Q}}(J) \cong \Omega^1_{C/\mathbb{Q}}(C).$$

Under mild conditions (e.g. $C(\mathbb{Q}) \neq \emptyset$), this generalises to

$$\Omega^1_{\mathcal{J}/\mathbb{Z}_{(p)}}(\mathcal{J}) \cong \omega_{\mathcal{C}/\mathbb{Z}_{(p)}}(\mathcal{C}),$$

where $\mathcal{J}/\mathbb{Z}_{(p)}$ is a Néron model of the Jacobian, and $\omega_{\mathcal{C}/\mathbb{Z}_{(p)}}$ is the canonical sheaf. Now our goal is to explicitly find generators of the $\mathbb{Z}_{(p)}$-module $\omega_{\mathcal{C}/\mathbb{Z}_{(p)}}(\mathcal{C})$.

Universiteit Leiden

How to calculate the real period P_J? (4/5)

We know that $\omega_{\mathcal{C}/\mathbb{Z}_{(p)}}(\mathcal{C}) \otimes_{\mathbb{Z}_{(p)}} \mathbb{Q} \cong \Omega^1_{C/\mathbb{Q}}(C)$, but how do we make this explicit?

Proposition

Let $\mathcal{C}/\mathbb{Z}_{(p)}$ be an affine curve, given inside $\mathbb{A}^n$ by the equations $f_2 = f_3 = \ldots = f_n = 0$. Suppose that $\mathcal{C}_\eta/\mathbb{Q}$ is smooth and that $\tau = g \cdot dx_1$ is a regular differential on $\mathcal{C}_\eta$.

Then in the canonical sheaf $\omega_{\mathcal{C}/\mathbb{Z}_{(p)}}$, the differential τ corresponds to a regular differential if and only if

$$\det \left(\frac{\partial f_i}{\partial x_j} \right)_{i,j=2}^{n} \cdot g$$

is regular in $\mathcal{O}_\mathcal{C}$.

How to calculate the real period P_J? (5/5)

This should not be confused with the sheaf of relative differentials.

Example: let $p > 2$ and $\mathcal{C} : f := y^2 - p(x^3 + 1) = 0$ and $\tau = \frac{dx}{y}$.

At first sight, it might seem that τ has a pole at the special fibre $p = 0$. However, inside the canonical sheaf, τ is

$$\frac{\partial f}{\partial y} \cdot \frac{1}{y} = 2$$

times a generator. As $p > 2$, this is a unit, and τ does not have a pole at the special fibre.

Ideas for the future

In the future, we hope to extend these methods to numerically verify (possibly up to squares) BSD for some smooth plane quartics.

Moreover, the algrithm could likely be improved drastically by using the new algorithm for the regular model by Dokchitser et al., and using the new method by Sutherland for the L-function.