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)

28 March 2017

Universiteit Leiden

Outline

In Magma the author 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

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

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

to predict the size of $\text{Ш}(J)$.

List of results

The algorithm 'confirmed' BSD for:

- all elliptic curves $y^2 = x^3 + ax + b$ with $a, b \in \{-15, \dots, 15\}$, 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 $a, b, c, d, e \in \{-10, \dots, 10\}$ 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 $a, b, c, d, e, f, g \in \{-3, \dots, 3\}$ 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.

Universiteit Leiden

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;
- for genus 3: some examples for which the conductor was too big, which prolongs the calculation of the L-function and the period;
- the curve $x^5 - 4x^4 + 8x^3 - 8x^2 + 4x - 1$ for which the height code takes too long to excute, for reasons still unknown to the author;
- the curve $x^5 - 3x^4 + 6x^3 - 6x^2 + 4^x - 1$ for which the L-function code takes too long to execute, for reasons still unkown to the author.

Universiteit Leiden

Runtimes

We recorded the following runtimes. Here H_1 is one of the curves from 'Empirical evidence' paper with $\text{Ш} \neq 1$, conductor 125 and $\Delta(H_1) = 5^{16}$. Moreover, H_2 is of genus 2 with $\Delta(H_2) = 62720$, and H_3 is of genus 3 with $\Delta(H_3) = -1523712$.

	H_1 (rk 0)	H_2 (rk 1)	H_3 (rk 1)
$\lim_{s\to1}(s-1)^{-r}L(J,s)$	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

Runtime in seconds

Universiteit Leiden

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

The *algebraic rank* of J can be computed by `Magma`, by computing an upper bound, using 2-Selmer groups, and a lower bound, by looking for points. In practice, these bounds seemed to agree for our examples.

The *L-function* and its derivatives can be evaluated at $s = 1$ using code by Tim Dokchitser. This routine uses the `RegularModel` routine in `Magma`.

Problem: the runtime seems to increase quickly as the regulator increases, as it uses the functional equation for the evaluation.

How to calculate the real period P_J? (1/2)

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. To correct for this, we need to find how far it is away from being a Néron differential.

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

For the primes of good reduction, it is alright, but for the primes p of bad reduction 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 and adjust the basis if necessary.

Problem: step 3 takes a lot of time: for p^g differentials a non-trivial calculation had to be done. It should not be too hard to overcome this problem.

Universiteit Leiden

How to calculate the regulator R_J?

We calculate the *algebraic rank* as before and try to find generators for the free part of $J(\mathbb{Q})_{\mathrm{tors}}$.

For higher genus hyperelliptic curves, the height pairing can then be calculated using techniques from Arakelov theory. The algorithm and its implementation are due to Holmes and Müller.

Problem: the bounds up to which height we need to find all points might be a bit big. In practice, the heights needed seem to be very small. In theory, it might happen that we get an error factor, which is a rational square.

Problem: the higher the genus gets, the harder it is find all points of low height. For genus 2, it is very doable, for higher genus it gets impractical. For genus 3, we at least always found a subgroup of full rank.

How to calculate the Tamagawa numbers c_p?

To calculate the Tamagawa number for a bad prime p, we compute a regular model $\mathcal{C}/\mathbb{Z}_{(p)}$ of C. The `Magma` code gives an explicit respresentation with charts and equations, which is a bit cumbersome to access.

We calculate the action of the absolute Galois group on the component group, by calculating the action of the Frobenius on the explicit equations.

Then the Tamagawa number is the size of the Galois-invariant subgroup of the component group.

Universiteit Leiden

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

Here we combine the following two approaches. For genus 2, this is already implemented in `Magma`.

As before, we calculate the points in $J(\mathbb{Q})$ up to a certain height and look for torsion points. This gives a lower bound for the torsion subgroup.

We calculate the reduction J_p of J modulo a bunch of small primes p of good reduction. The prime-to-p-part of the torsion group injects into $J_p(\mathbb{F}_p)$. This gives an upper bound.

In most cases, the upper and lower bound coincided. In those cases it did not (genus 3), this might induce an error factor equal to a rational square in the BSD formula.

Universiteit Leiden

Ideas for the future

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

Right now, the verification cannot be done for a lot of curves, because `Magma` cannot compute a regular model. This happens when a component in the special fibre has to be blown up and this component is not defined over $\mathbb{F}_p$.

Another idea could be to extend the `RegularModel` code to also cope with these cases. The author did not extensively research the feasibility of this suggestion.

Universiteit Leiden

The end

Questions / comments / discussion ?