Computing Hilbert class polynomials

Juliana Belding, University of Maryland

Reinier Bröker, Microsoft Research

Andreas Enge, École Polytechnique

Kristin Lauter, Microsoft Research

ANTS VIII, Banff
May 2008

Hilbert class polynomial

Throughout this talk, $D < 0$ is a discriminant. Let $\mathcal{O}_D$ be *the* imaginary quadratic order of discriminant D.

The quotient $\mathbf{C}/\mathcal{O}_D$ has a natural structure of an *elliptic curve*.

The *Hilbert class polynomial* P_D is the minimal polynomial over $\mathbf{Q}$ of the j-value $j(\mathbf{C}/\mathcal{O}_D)$. It defines the *ring class field* H of $\mathcal{O}_D$.

Non-trivial fact: $P_D \in \mathbf{Z}[X]$.

Example.

$P_{-23} = X^3 + 3491750X^2 - 5151296875X + 12771880859375.$

Classical algorithm to compute P_D

- list reduced binary quadratic forms $aX^2+bXY+cY^2$ of discriminant $b^2-4ac=D$
- compute

$$P_D = \prod_{[a,b,c]} \left(X - j\left(\frac{-b+\sqrt{D}}{2a} \right) \right) \in \mathbf{Z}[X].$$

Here j is the complex analytic modular function $\mathbf{H} \to \mathbf{C}$ with Fourier expansion $j(z) = 1/q + 744 + 196884q + \ldots$ in $q = \exp(2\pi i z)$.

We compute $j(\frac{-b+\sqrt{D}}{2a}) \in \mathbf{C}$ with high enough accuracy to be able to round the coefficients of the expanded product to the nearest integer.

Run time (Enge): $\widetilde{O}(|D|)$. Rounding errors might occur.

Second approach

p-adic approach (Couveignes-Henocq (2002), Bröker (2006))

1. Find a small prime p that splits in H.

2. Find an elliptic curve $E/\mathbf{F}_p$ with $\mathrm{End}(E) = \mathcal{O}_D$.

3. Lift $j(E) \in \mathbf{F}_p$ to its canonical lift $\widetilde{j(E)} \in \mathbf{Q}_p$. We have $P_D(\widetilde{j(E)}) = 0$.

4. Compute the Galois conjugates of $\widetilde{j(E)}$.

5. Expand $P_D = \prod_{\mathfrak{a} \in \mathrm{Cl}(\mathcal{O}_D)} (X - \widetilde{j(E)}^{\mathfrak{a}}) \in \mathbf{Z}[X]$.

Run time: $\widetilde{O}(|D|)$ under GRH. No rigorous bound.

Third approach

If we have bounds on an integer $x \in \mathbf{Z}$, we can 'reconstruct' it using the Chinese remainder theorem.

Example: only positive integer $x \leq 1000$ with $x \equiv 5 \bmod 7$, $x \equiv 8 \bmod 11$ and $x \equiv 0 \bmod 13$ is $x = 481$.

Idea is used in Schoof's point counting algorithm.

Lauter et al.: compute $P_D \in \mathbf{F}_p[X]$ for various primes p, and then apply Chinese remaindering.

Their run time: $O(|D|^{3/2+o(1)})$.

Today. Change their algorithm to obtain $\widetilde{O}(|D|)$ as well. Analyze the log-factors.

Size of the output

The degree of P_D equals the class number $h(D)$.

For z in the fundamental domain of $\mathrm{SL}_2(\mathbf{Z})\backslash\mathbf{H}$, we have $|j(z) - q^{-1}| \leq 2150$. Hence: $j(z) \approx q^{-1}$.

The largest coefficient of P_D has size bounded by

$$3h(D) + \pi\sqrt{|D|} \sum_{[a,b,c]} \frac{1}{a}.$$

Unconditional bound (Schur, 1918): $h(D) = O(|D|^{1/2} \log |D|)$.

GRH bound (Littlewood, 1928): $h(D) = O(|D|^{1/2} \log\log |D|)$.

Size of the output

'Classical' bound (Schoof, 1991): $\sum_{[a,b,c]} 1/a = O((\log|D|)^2)$.

New technique (based on idea of Granville and Stark):

$$\sum_{[a,b,c]} 1/a \leq \sum_{a \leq \sqrt{|D|}} \frac{\prod_{p|a}\left(1 + \left(\frac{D}{p}\right)\right)}{a}.$$

Take the Euler product to get the bound

$$\prod_{p \leq \sqrt{|D|}} \left(1 + \frac{1}{p}\right)\left(1 + \frac{\left(\frac{D}{p}\right)}{p}\right) \leq c \log|D| \prod_{p \leq \sqrt{|D|}} \frac{1}{1 - \left(\frac{D}{p}\right)/p}.$$

GRH-bound: $\sum_{[a,b,c]} 1/a = O(\log|D| \log\log|D|)$.

Size of the primes

Bound on size of largest coefficient of P_D is

$$B_D = O(|D|^{1/2} \log |D| \log\log |D|).$$

We can use a prime p if and only if p splits completely in H.

Need *many* primes p, the product should exceed B_D.

Effective Chebotarev. (under GRH) *For the smallest p we have*

$$p = O(|D|(\log |D|)^4).$$

All the $O(|D|^{1/2} \log\log |D|)$ primes we need satisfy this bound.

Computing $P_D \in \mathbf{F}_p[X]$

Fix a prime p that splits completely in the ring class field H.

Step 1. Find a curve $E/\mathbf{F}_p$ with $\mathrm{End}(E) = \mathcal{O}_D$.

Step 2. Compute the Galois conjugates $j(E)^{\mathfrak{a}}$ for $\mathfrak{a} \in \mathrm{Cl}(\mathcal{O}_D)$.

Step 3. Return $P_D \bmod p = \prod_{\mathfrak{a} \in \mathrm{Cl}(\mathcal{O}_D)} (X - j(E)^{\mathfrak{a}}) \in \mathbf{F}_p[X]$.

Step 1: find a curve with the right endomorphism ring

Write $4p = x^2 - Dy^2$. The curves we are looking for have

$$p + 1 \pm x$$

points over $\mathbf{F}_p$. Reason: $\mathcal{O}_{x^2-4p} \subseteq \mathcal{O}_D$.

Naïve algorithm

Find a curve $E/\mathbf{F}_p$ with $p + 1 \pm x$ points by *trying random* curves. Count the number of points of each 'test curve'.

This suffices for the overall $\widetilde{O}(|D|)$ runtime. However: point counting has the 'slow' runtime $\widetilde{O}((\log p)^5)$.

Step 1a: instead of counting points

Pick a random point $P \in E(\mathbf{F}_p)$ and see if $(p+1\pm x)P$ holds. If not, try the 'next' curve.

Select a few random points on E and its quadratic twist E' and compute their orders *assuming* they divide $p+1\pm x$. If this fails, try the 'next' curve.

We have $E(\mathbf{F}_p) \cong \mathbf{Z}/n_1\mathbf{Z} \times \mathbf{Z}/n_2\mathbf{Z}$ with $n_1 \mid n_2$. A fraction $\varphi(n_2)/n_2$ of all points have *maximal order*. We quickly find points P, P' of maximal order.

Mestre: for $p > 457$ either $\mathrm{ord}(P) \geq 4\sqrt{p}$ or $\mathrm{ord}(P') \geq 4\sqrt{p}$.

Runtime drops to $\widetilde{O}((\log p)^3)$.

Step 1: finding a curve with the right endomorphism ring

Let $E/\mathbf{F}_p$ have $p + 1 \pm x$ points.

Compute $\mathrm{End}(E) = \mathcal{O}_{Dn^2}$ using Kohel's algorithm.

If $\mathrm{End}(E) = \mathcal{O}$, we are done. Otherwise: find another random curve with $p + 1 \pm x$ points.

Run time. Domininated by the first step.

GRH $\Longrightarrow O((p/h(D))(\log p)^{3+o(1)}) = O(|D|^{1/2}(\log |D|)^{7+o(1)})$.

Step 2: computing the Galois conjugates

The class group $\mathrm{Cl}(\mathcal{O}_D)$ acts on the set of elliptic curves with endomorphism ring $\mathcal{O}_D$ via

$$j(E) \mapsto j(E)^I \stackrel{\text{def}}{=} j(E/E[I]) \qquad \text{for } [I] \in \mathrm{Cl}(\mathcal{O}_D).$$

The value $j(E)^I$ is a root of $\Phi_l(j(E), X) \in \mathbf{F}_p[X]$ for I of prime norm l. Here: Φ_l is the l-th modular polynomial.

Can prove: under 'harmless assumptions': only two roots in $\mathbf{F}_p$:

$$j(E)^I \qquad \text{and} \qquad j(E)^{\overline{I}}.$$

We need both.

Step 2: computing the Galois conjugates

(*Bach bound / effective Chebotarev*):
GRH $\Longrightarrow$ $\mathrm{Cl}(\mathcal{O}_D)$ is generated by ideals of norm $O((\log|D|)^2)$.

Finding a Galois conjugate of $j(E)$ takes time $\widetilde{O}((\log|D|)^5)$.
There are $h(D) \overset{\mathrm{GRH}}{=} O(|D|^{1/2}\log\log|D|)$ conjugates.

Time to find all Galois conjugates is $O(|D|^{1/2}(\log|D|)^{5+o(1)})$.

Total time spent so far:

$$O(|D|^{1/2}(\log|D|)^{7+o(1)}).$$

This dominates Step 3: expanding the product.

Computing P_D, conclusion

Time per prime p: $O(|D|^{1/2}(\log|D|)^{7+o(1)})$.

We need $O(|D|^{1/2}\log\log|D|)$ primes. Total time:

$$O(|D|(\log|D|)^{7+o(1)}).$$

Recombining using 'classical' Chinese remaindering would take too much time.

Fast Chinese remaindering ('fancy product trees') takes time $O(|D|^{1/2}(\log|D|)^{3+o(1)})$ per coefficient.

We have $h(D) \overset{\text{GRH}}{=} O(|D|^{1/2}\log\log|D|)$ coefficients.

Run time for entire algorithm: $O(|D|(\log|D|)^{7+o(1)})$, under GRH.

Comparison

Complex analytic.	$O(\lvert D\rvert(\log\lvert D\rvert)^{5+o(1)})$	*rigorous*
possible rounding errors	$O(\lvert D\rvert(\log\lvert D\rvert)^{3+o(1)})$	*GRH*
p**-adic.**	$O(\lvert D\rvert(\log\lvert D\rvert)^{6+o(1)})$	*GRH*
	$O(\lvert D\rvert(\log\lvert D\rvert)^{3+o(1)})$	*heuristic*
CRT.	$O(\lvert D\rvert(\log\lvert D\rvert)^{7+o(1)})$	*GRH*
	?	*heuristic*

Heuristics for CRT

Size of primes p is 'pessimistic'.

We look for solutions to $x^2 - Dy^2 = 4p$. Reason: p splits completely in H iff p splits in principal primes in $\mathcal{O}_D$.

To find solutions, let x, y range over $1, 2, \ldots$ until we find a solution with p prime.

Heuristics: one out of every $\log |D|$ integers around $|D|$ is prime.

Size of primes becomes: $O(|D| \log |D|)$ instead of $O(|D|(\log |D|)^4)$.

Heuristics for CRT

Bottlenecks in run time: finding $E/\mathbf{F}_p$ with $p+1\pm x$ points and size of generators of $\mathrm{Cl}(\mathcal{O}_D)$.

1. Instead of computing *orders* of random points P, only check if $(p+1\pm x)P = 0_E$ holds.

2. People 'believe': $\mathrm{Cl}(\mathcal{O}_D)$ is generated by primes of size $\widetilde{O}(\log|D|)$.

Heuristic run time becomes: $O(|D|(\log|D|)^{3+o(1)})$.

One of the bottlenecks is now Chinese remaindering!

Comparison

Complex analytic.	$O(\lvert D\rvert(\log\lvert D\rvert)^{5+o(1)})$	*rigorous*
possible rounding errors	$O(\lvert D\rvert(\log\lvert D\rvert)^{3+o(1)})$	*GRH*
p**-adic.**	$O(\lvert D\rvert(\log\lvert D\rvert)^{6+o(1)})$	*GRH*
	$O(\lvert D\rvert(\log\lvert D\rvert)^{3+o(1)})$	*heuristic*
CRT.	$O(\lvert D\rvert(\log\lvert D\rvert)^{7+o(1)})$	*GRH*
	$O(\lvert D\rvert(\log\lvert D\rvert)^{3+o(1)})$	*heuristic*

Practical performance

CRT-approach appears to be slow in practice. Reason: we need many ‘large’ primes.

To speed it up: use *inert* primes.

Easiest case: $D \equiv 5 \bmod 8 \Longrightarrow P_D \bmod 2 = X^{h(D)}$.

We can compute $P_D \bmod p$ for *any* inert prime, see ANTS-article.

Run time is very bad with respect to p. Needs to be analyzed how much this will speed up the method.

As it stands now: the complex analytic method is the fastest in practice.

Proving the exponent 3?

We probably cannot do better than $O(|D|(\log|D|)^{3+o(1)})$: this is the time it takes to expand $\prod_j (X-j)$.

Question. *Can we prove (under GRH) this run time without the rounding error problem?*

Answer? Use p-adic lifting for an *inert* prime p.

See ANTS-article for the $p \equiv 1 \bmod 12$ algorithm and the PhD-thesis of Juliana Belding for general p.

Run time analysis: ≥ 2008.