Computing modular polynomials

Reinier Bröker

BROWN

joint work with Kristin Lauter (Microsoft Research)
and Drew Sutherland (MIT)

Modular polynomials

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

Definition. *For $m > 1$, the modular polynomial Φ_m is the minimal polynomial of $j(mz)$ over $\mathbf{C}(j)$.*

Properties.

- $\Phi_m(X) \in \mathbf{Z}[X, j]$;
- $\Phi_m(X, j) = \Phi_m(j, X)$;
- m prime $\Longrightarrow \deg_X(\Phi_m) = m + 1$.

An example

$$
\begin{aligned}
\Phi_5 = {} & X^6 - X^5Y^5 + 3720X^5Y^4 - 4550940X^5Y^3 + 2028551200X^5Y^2 \\
& -246683410950X^5Y + 1963211489280X^5 + 3720X^4Y^5 \\
& +1665999364600X^4Y^4 + 107878928185336800X^4Y^3 \\
& +383083609779811215375X^4Y^2 + 128541798906828816384000X^4Y \\
& +1284733132841424456253440X^4 - 4550940X^3Y^5 \\
& +107878928185336800X^3Y^4 - 441206965512914835246100X^3Y^3 \\
& +26898488858380731577417728000X^3Y^2 \\
& -192457934618928299655108231168000X^3Y \\
& +280244777828439527804321565297868800X^3 \\
& \vdots \\
& +53274330803424425450420160273356509151232000X + Y^6 \\
& +1963211489280Y^5 + 1284733132841424456253440Y^4 \\
& +280244777828439527804321565297868800Y^3 \\
& +6692500042627997708487149415015068467200Y^2 \\
& +53274330803424425450420160273356509151232000Y \\
& +141359947154721358697753474691071362751004672000
\end{aligned}
$$

Why compute it?

The modular polynomial Φ_m is a model for the modular curve $Y_0(m)$ parametrizing elliptic curves that are m-isogenous.

In particular: E_1, E_2 are m-isogenous $\iff \Phi_m(j(E_1), j(E_2)) = 0$.

This *moduli interpretation* is valid over all fields of characteristic coprime to m. (The curve $Y_0(m)$ has good reduction modulo $p \nmid m$.)

Over $\mathbf{F}_p$, the polynomials Φ_m are used for *point counting*, *endomorphism ring computations*, *cryptography*, etc., etc.

In various algorithms, computing this 'building block' Φ_m for moderately large m is actually a *bottleneck*.

Size of Φ_m

The polynomial Φ_m is *big*: it has size $\widetilde{O}(m^3)$.

No useful lower bound is known, and $m^{3+\varepsilon}$ appears to be the 'true size'.

m	coefficients	largest	average	total
127	8258	7.5kb	5.3kb	5.5MB
251	31880	16kb	12kb	48MB
503	127262	36kb	27kb	431MB
1009	510557	78kb	60kb	3.9GB
2003	2009012	166kb	132kb	33GB
3001	4507505	259kb	208kb	117GB
4001	8010005	356kb	287kb	287GB
5003	12522512	454kb	369kb	577GB
10007	50085038	968kb	774kb	4.8TB

Previous algorithms to compute Φ_l

- linear algebra on the q-expansions of $j(z)$ and $j(lz)$.
  - ◇ Atkin was the first ($\leq$ 1992), after him many people.
  - ◇ run time: $O(l^4(\log l)^{3+\varepsilon})$

- use Vélu's formulas to write down isogenies
  - ◇ Charles, Lauter (2005)
  - ◇ run time: $O(l^{5+\varepsilon})$

- evaluation-interpolation of complex functions
  - ◇ Enge (2009)
  - ◇ run time: $O(l^3(\log l)^{4+\varepsilon})$
  - ◇ almost *optimal* run time! This algorithm broke all world records a year ago.

New result

We compute Φ_l modulo carefully selected primes p and combine the results using the Chinese remainder theorem.

Computing Φ_l mod p takes time $O(l^2(\log p)^{3+\varepsilon})$.

If GRH holds true, we can find enough small primes p. We compute $\Phi_l \in \mathbf{Z}[X, Y]$ in time

$$O(l^3(\log l)^{3+\varepsilon}).$$

Performance highlights.

- $l = 251$: 40 seconds (old record: 688 seconds)
- $l = 1009$: 3822 seconds (old record: 107200 seconds)

Our algorithm computes Φ_l at 'a rate of 1 MB/s'.

Computing Φ_l mod p

Given a prime $l > 2$, *fix* an imaginary quadratic order $\mathcal{O}$ satisfying

- $\mathcal{O}$ is maximal at l;
- $h(\mathcal{O}) \geq l + 2$.

Example: for $l > 3$ take an order of large enough 3-power index in $\mathbf{Q}(\sqrt{-7})$.

We will compute Φ_l mod p for primes p that

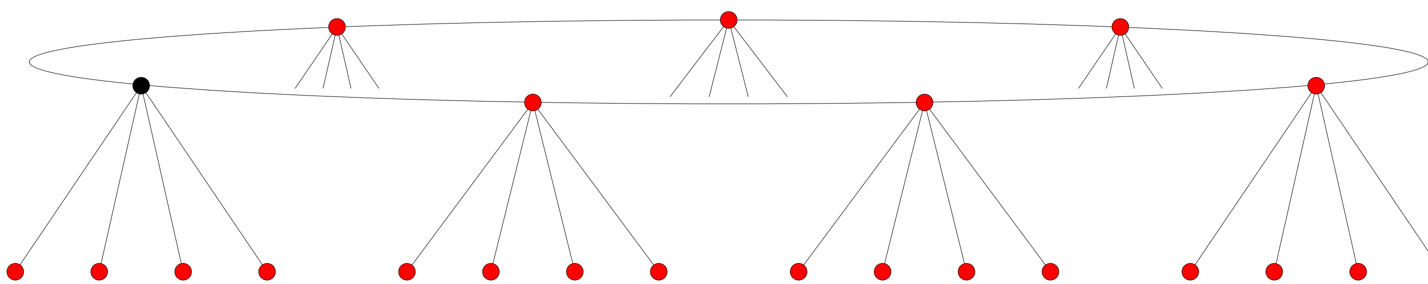
- split completely in the ray class field of conductor l for $\mathcal{O}$;
- do *not* split complely in the ring class field for $\mathbf{Z} + l^2\mathcal{O}$.

If GRH holds true, there are many primes p of size $\log p = O(\log l)$ satisfying these conditions.

Elliptic curves over $\mathbf{F}_p$

By construction, all elliptic curves with endomorphism ring $\mathcal{O}$ are defined over $\mathbf{F}_p$ *and* their l-torsion points live over $\mathbf{F}_p$.

All elliptic curves with endomorphism ring $\mathbf{Z} + l\mathcal{O}$ also live over $\mathbf{F}_p$, but *none* of their non-trivial l-torsion subgroups are defined over $\mathbf{F}_p$.



Strategy.

- Find the black point $j(E)$ by computing a root of $H_{\mathcal{O}}$ mod p.
- Find its $l + 1$ neighbors.
- Compute $\Phi_l(j(E), X) = \prod_{\text{neighbors } E' \text{ of } E} (X - j(E')) \in \mathbf{F}_p[X]$.

Finding the neighbors in case l splits

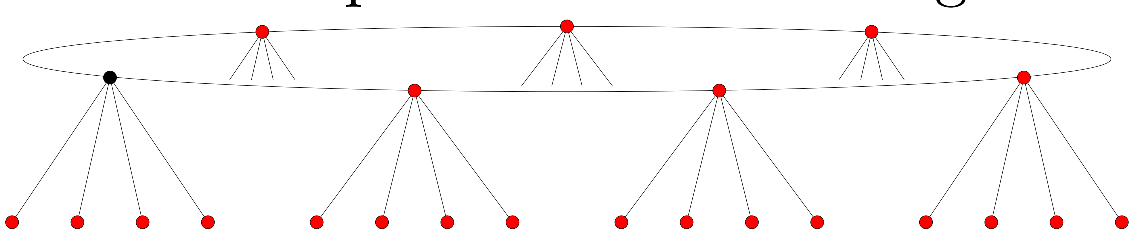
The 2 neighbors 'at the surface' are $j(E/E[\mathfrak{l}_1])$ and $j(E/E[\mathfrak{l}_2])$ with $(l) = \mathfrak{l}_1\mathfrak{l}_2 \subset \mathcal{O}$.

Observation: if $[\mathfrak{l}_1] = [I] \in \mathrm{Pic}(\mathcal{O})$, then $j(E/E[\mathfrak{l}_1]) = j(E/E[I])$. Since we pick $\mathcal{O}$ ourselves, I can be chosen to be very *smooth*.

For instance, for the order $\mathcal{O}$ of index 3^n in $\mathbf{Q}(\sqrt{-7})$ we get

$$\mathrm{Pic}(\mathcal{O}) = \mathbf{Z}/(4 \times 3^{n-1}\mathbf{Z}) = \langle \mathfrak{p}_2 \rangle$$

and we only have to compute a series of 2-isogenies.



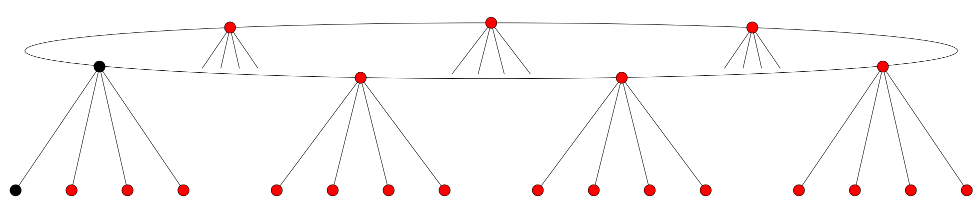
Finding the neighbors in case l splits

To find the other $l-1$ neighbors, we could use Vélu's formulas. However: this turns out to be too slow.

Instead: use Vélu *once* to find one neighbor $j(E_1)$ and use the following.

Lemma. *Write* $R = \mathbf{Z} + l\mathcal{O}$. *The kernel of* $\mathrm{Pic}(R) \xrightarrow{\varphi} \mathrm{Pic}(\mathcal{O})$ *is generated by an invertible* R*-ideal* J *of norm* l^2.

Proof. Look at the $l+1$ index l subrings of $\mathcal{O}$. We find the $\mathcal{O}$-ideals of norm l, the ring R, and the others are fractional invertible R-ideals J_i of norm l^2. Now observe $\varphi(J_i) = l\mathcal{O}$. □



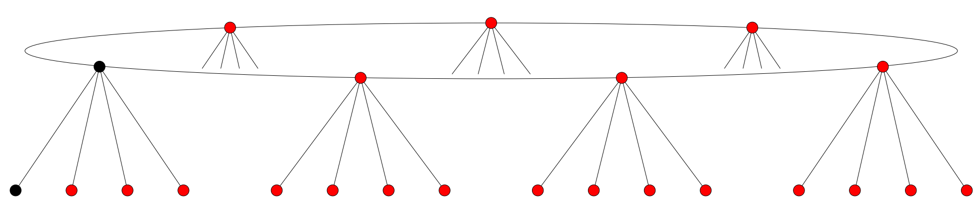
Finding the neighbors in case l splits

We have $\mathrm{Ker}(\varphi) = \langle [J] \rangle \subseteq \mathrm{Pic}(R)$.

Simplest case: $\mathrm{Pic}(R)$ is cyclic. Again, we may assume that it is generated by an ideal of *small* norm.

Write $[J] = [\mathfrak{q}^n]$. Compute the action of $\mathfrak{q}, \mathfrak{q}^2, \mathfrak{q}^3, \ldots, \mathfrak{q}^{h(R)}$ on the black point $j(E_1)$ at the 'floor'.

The points $j(E_1/E_1[\mathfrak{q}^n])$, $j(E_1/E_1[\mathfrak{q}^{2n}])$, $\ldots$ are the neighbors we are looking for.



Repeating this procedure

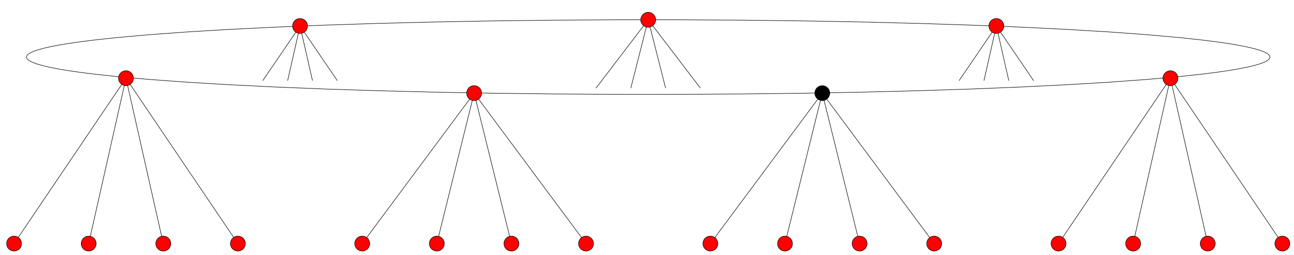
Once we have computed

$$\Phi_l(j(E), X) = \prod_{\text{neighbors } E' \text{ of } E} (X - j(E')) \in \mathbf{F}_p[X],$$

we need to pick another point on 'the surface' and repeat everything.

However: this will require *much less* work.

Reason. All its neighbors on the floor have already been computed! This is crucial to proving the run time and the practical performance.



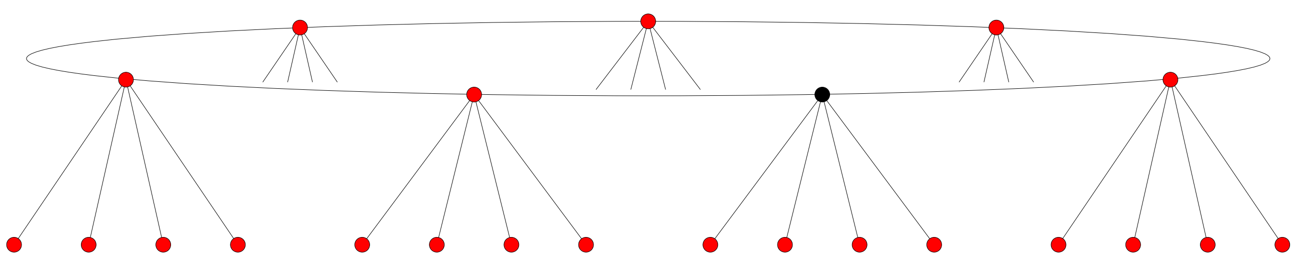
Interpolating

We need to compute $\Phi_l(j(E), X) \in \mathbf{F}_p[X]$ for $l + 2$ different values $j(E)$ that are 'on the surface'.

Next step is to *interpolate* $\Phi_l(X, Y) \in (\mathbf{F}_p[X])(Y)$.

We repeat this for *many* primes p until we can compute $\Phi_l(X, Y) \in \mathbf{Z}[X, Y]$.

Remark. We can compute $\Phi_l(X, Y)$ mod s without first computing it over $\mathbf{Z}$. This saves space!



Remarks about the proof of the run time

We need an *explicit height bound* on Φ_l to know 'when to stop'.

Paula Cohen (1984): $h(\Phi_l) = 6l \log l + O(l)$.
Bröker, Sutherland (2009): $h(\Phi_l) = 6l \log l + 17l$.

Our 'example order' $\mathcal{O}$ can be used throughout the proof.

The proof needs GRH to ensure that the required 'small' splitting primes p exist. To bound their sizes, use effective Chebotarev, Hasse's *Führerdiskriminantenproduktformel*, etc.

To bound the time for the interpolation, use results from computer science.

We have computed Φ_l for all primes $l \leq 3607$.