About the CRT method to compute class polynomials in dimension 2

Séminaire LFANT

Kristin Lauter[1], **Damien Robert**[2]

[1]Microsoft Research [2]LFANT Team, IMB & INRIA Bordeaux Sud-Ouest

10/05/2012 (Bordeaux)

Motivation

Abelian varieties and cryptography

If $A/\mathbb{F}_q$ is a "generic" abelian variety of small dimension g, then the DLP on $A(\mathbb{F}_q)$ is thought to be hard if $\#A(\mathbb{F}_q)$ is divisible by a large prime.

- Take random abelian varieties and count the number of points (a bit too slow when $g = 2$);
- Generate abelian varieties with a prescribed number of points ($\Rightarrow$ paring based cryptography).

Class polynomials

- If $A/\mathbb{F}_q$ is an ordinary (simple) abelian variety of dimension g, $\mathrm{End}(A)\otimes\mathbb{Q}$ is a (primitive) CM field K (K is a totally imaginary quadratic extension of a totally real number field K_0).
- The class polynomials $H_1, \widehat{H}_2 \dots, \widehat{H}_{g(g+1)/2}$ parametrizes the invariants of all abelian varieties $A/\mathbb{C}$ with $\mathrm{End}(A) \simeq O_K$.
- If the class polynomials are totally split modulo $\mathfrak{P}$, their roots in $\mathbb{F}_{\mathfrak{P}}$ gives invariants of abelian varieties $A/\mathbb{F}_{\mathfrak{P}}$ with $\mathrm{End}(A) \simeq O_K$. It is easy to recover $\#A(\mathbb{F}_{\mathfrak{P}})$ given O_K and $\mathfrak{P}$.

Some technical details

- The abelian varieties are principally polarized.
- A CM type Φ is a choice of an extension to K for each of the embedding $K_0 \to \mathbb{R}$. We have

$$\mathrm{Hom}(K,\mathbb{C}) = \Phi \oplus \overline{\Phi}.$$

  Example: If K is a (primitive) CM field of degree 4, then either K is cyclic and there is one class of CM type, or K is dihedral and there is two class of CM types.
- If A is an abelian variety with CM by K, the representation $K \to \mathrm{End}\, T_0 A$ is given by a CM type Φ.
- The isogeny class of complex abelian varieties with CM by K is determined by the class of Φ.
- The reflex field of (K,φ) is the CM field K^r generated by the traces $\sum_{\varphi \in \Phi} \varphi(x)$, $x \in K$.
- The type norm $N_\Phi : K \to K^r$ is $x \mapsto \prod_{\varphi \in \Phi} \varphi(x)$.

Definition

The class polynomials $(H_\Phi)_i$) parametrizes the abelian varieties with CM by (O_K,Φ)

Class polynomials and complex multiplication

Theorem (Main theorems of complex multiplication)

- *The class polynomials $(H_\Phi)_i$ are defined over K_0 and generate a subfield $\mathfrak{H}_\Phi$ of the Hilbert class field of K^r.*
- *If $A/\mathbb{C}$ has CM by (O_K, Φ) and $\mathfrak{P}$ is a prime of good reduction in $\mathfrak{H}_\Phi$, then the Frobenius of $A_{\mathfrak{P}}$ corresponds to $N_{\mathfrak{H}_\Phi, \Phi^r}(\mathfrak{P})$.*

If $g \leqslant 2$, the CM types are in the same orbits under the absolute Galois action, and the class polynomials $H_i = \prod_\Phi (H_\Phi)_i$ are rationals (and even integrals when $g = 1$).

- For efficiency, we compute the class polynomials H_Φ since they give a factor of the full class polynomials H. This mean we need less precision.
- In genus 2, this involves working over K_0 rather than $\mathbb{Q}$ in the Dihedral case.

Constructing class polynomials

- Analytic method: compute the invariants in $\mathbb{C}$ with sufficient precision to recover the class polynomials.
- p-adic lifting: lift the invariants in $\mathbb{Q}_p$ with sufficient precision to recover the class polynomials (require specific splitting behavior of p).
- CRT: compute the class polynomials modulo small primes, and use the CRT to reconstruct the class polynomials.

Remark

In genus 1, all these methods are quasi-linear in the size of the output $\Rightarrow$ computation bounded by memory. But we can construct directly the class polynomials modulo p with the explicit CRT.

Review of the CRT algorithm in genus 2

1. Select a CRT prime p.
2. For each abelian surface A in the $O(p^3)$ isomorphic classes:
   2.1 Check if A is in the right isogeny class by computing the characteristic polynomial of the Frobenius (do some trial tests to check for $\#A$ before).
   2.2 Check if $\mathrm{End}(A) = O_K$.
3. From the invariants of the maximal curves, reconstruct $(H_\Phi)_i$ mod p.

Repeat until we can recover $(H_\Phi)_i$ from the $(H_\Phi)_i$ mod p using the CRT.

Remark

Since K is primitive, we only need to look at Jacobians of hyperelliptic curves of genus 2.

Selecting the prime p

Definition

A CRT prime $\mathfrak{p} \subset O_{K_0^r}$ is a prime such that all abelian varieties over $\mathbb{C}$ with CM by (O_K, Φ) have good reduction modulo $\mathfrak{p}$.

- $\mathfrak{p}$ is a CRT prime for the CM type Φ if and only if there exists an unramified prime $\mathfrak{q}$ in O_{K^r} of degree 1 above p of principal type norm (π)
- The isogeny class of the reduction of these abelian varieties mod $\mathfrak{p}$ is determined (up to a twist) by $\pm\pi$ where $N_\Phi(\mathfrak{p}) = (\pi)$.
- For efficiency, we work with CRT primes $\mathfrak{p}$ that are unramified of degree one over $p = \mathfrak{p} \cap \mathbb{Z}$.

$\Rightarrow$ the reduction to $\mathbb{F}_p$ of the abelian varieties with CM by (O_K, Φ) will then be ordinary.

Working with both CM types in the Dihedral case

Let Φ_1 and Φ_2 be the two CM types.

- If p splits as $\mathfrak{p}_1\mathfrak{p}_2$ in K_0^r, then for p to be a CRT prime for both CM types, we need $\mathfrak{p}_1$ and $\mathfrak{p}_2$ to be CRT primes.

$\Rightarrow$ We have less prime to work with, and less possibilities to sieve. Whereas when only dealing with one CM type, we can even choose the best prime among $\mathfrak{p}_1$ and $\mathfrak{p}_2$.

Remark

The reductions of the abelian varieties with CM by Φ_2 modulo $\mathfrak{p}_1$ are isomorphics to the reductions of the abelian varieties with CM by Φ_1 modulo $\mathfrak{p}_2$.

Checking if a curve is maximal

- Let J be the Jacobian of a curve in the right isogeny class. Then $\mathbb{Z}[\pi,\overline{\pi}] \subset \mathrm{End}(J) \subset O_K$.
- Let $\gamma \in O_K \backslash \mathbb{Z}[\pi,\overline{\pi}]$. We want to check if $\gamma \in \mathrm{End}(J)$.
- If $p > 3$ then $(O_K : \mathbb{Z}[\pi,\overline{\pi}])$ is prime to p. We then have $\gamma \in \mathrm{End}(J) \Leftrightarrow p\gamma \in \mathrm{End}(J)$.
- Let n be the smallest integer thus that $n\gamma \in \mathbb{Z}[\pi,\overline{\pi}]$. Since $(\mathbb{Z}[\pi,\overline{\pi}] : \mathbb{Z}[\pi]) = p$, we can write $np\gamma = P(\pi)$.
- Then $\gamma \in \mathrm{End}(J) \Leftrightarrow P(\pi) = 0$ on $J[n]$.
- In practice (Freeman-Lauter): compute $J[\ell^d]$ for $\ell^d \mid (O_K : \mathbb{Z}[\pi,\overline{\pi}])$ and check the action of the generators of O_K on it.

Remark

If $1, \alpha, \beta, \gamma$ are generators of O_K as a $\mathbb{Z}$-module, it can happen that $\gamma = P(\alpha, \beta)$, so that we don't need to check that $\gamma \in \mathrm{End}(J)$.

Example 1: Checking if a curve is maximal

- Let $H: y^2 = 10x^6 + 57x^5 + 18x^4 + 11x^3 + 38x^2 + 12x + 31$ over $\mathbb{F}_{59}$ and J the Jacobian of H. We have $\mathrm{End}(J) \otimes \mathbb{Q} = \mathbb{Q}(i\sqrt{29 + 2\sqrt{29}})$ and we want to check if $\mathrm{End}(J) = O_K$.
- O_K is generated as a $\mathbb{Z}$-module by $1, \alpha, \beta, \gamma$. α is of index 2 in $O_K / \mathbb{Z}[\pi, \overline{\pi}]$, β of index 4 and γ of index 40.
- So the old algorithm will check $J[2^3]$ and $J[5]$.
- But $(O_K)_2 = \mathbb{Z}_2[\pi, \overline{\pi}, \alpha]$, so we only need to check $J[2]$ and $J[5]$.

Field of definition of the ℓ^d-torsion

Proposition

- *The geometric points of* $J[\ell^d]$ *are defined over* $\mathbb{F}_{p^{\alpha_d}} \Leftrightarrow \pi^{\alpha_d} - 1 \in \ell^d \operatorname{End}(J)$.
- $\alpha_d \mid \alpha_1 \ell^{d-1}$. *If* $\operatorname{End}(J) = O_K$ *this is an equality:* $\alpha_d = \alpha_1 \ell^{d-1}$.

Corollary

Let α *be thus that* $\pi^\alpha - 1 \in \ell O_K$. *We first check that* $(\pi^\alpha - 1)/\ell$ *is an element of* $\operatorname{End}(J)$ *(*$\Leftrightarrow$ $J[\ell]$ *defined over* $\mathbb{F}_{p^\alpha}$*). Then* $J[\ell^d]$ *is defined over* $\mathbb{F}_{p^{\alpha \ell^{d-1}}}$.

Remark

It may happen that we get a factor two on the degrees by working over the twist: that is by working with $-\pi$.

Computing the ℓ^d-torsion

- We compute $\#J(\mathbb{F}_{p^\alpha}) = \ell^\beta c$ (where α is the degree of definition of the ℓ^d-torsion).
- If P_0 is a random point of $J(\mathbb{F}_{p^\alpha})$, then $P = cP_0$ is a random point of ℓ^∞-torsion, and P multiplied by a suitable power of ℓ is a random point of ℓ^d-torsion.
- Usual method (Freeman-Lauter): take a lot of random points of ℓ^d-torsion, and hope they generate it over $\mathbb{F}_{p^\alpha}$.
- Problems: the random points of ℓ^d-torsion are not uniform $\Rightarrow$ require a lot of random points, and the result is probabilistic.
- Our solution: Compute the whole ℓ^∞-torsion. "Correct" points to find uniform points of ℓ^d-torsion. Use pairings to save memory.

$\Rightarrow$ We can check if a curve is maximal faster.

$\Rightarrow$ We can abort early.

Example 2: checking if a curve is maximal

- Let $H : y^2 = 80x^6 + 51x^5 + 49x^4 + 3x^3 + 34x^2 + 40x + 12$ over $\mathbb{F}_{139}$ and J the Jacobian of H. We have $\mathrm{End}(J) \otimes \mathbb{Q} = \mathbb{Q}(i\sqrt{13 + 2\sqrt{29}})$ and we want to check if $\mathrm{End}(J) = O_K$.
- For that we need to compute $J[3^5]$, that lives over an extension of degree 81 (for the twist it lives over an extension of degree 162).
- With the old randomized algorithm, this computation takes 470 seconds (with 12 Frobenius trials over $\mathbb{F}_{139^{162}}$).
- With the new algorithm computing the ℓ^∞-torsion, it only takes 17.3 seconds (needing only 4 random points over $\mathbb{F}_{139^{81}}$, approx 4 seconds needed to get a new random point of ℓ^∞-torsion).

Obtaining all the maximal curves

- If J is a maximal curve, and ℓ does not divide $(O_K : \mathbb{Z}[\pi,\overline{\pi}])$, then any (ℓ,ℓ)-isogenous curve is maximal.
- The maximal Jacobians form a principal homogeneous space under the Shimura class group $\mathfrak{C}(O_K) = \{(I,\rho) \mid I\overline{I} = (\rho) \text{ and } \rho \in K_0^+\}$.
- (ℓ,ℓ)-isogenies between maximal Jacobians correspond to element of the form $(I,\ell) \in \mathfrak{C}(O_K)$. We can use the structure of $\mathfrak{C}(O_K)$ to determine the number of new curves we will obtain with (ℓ,ℓ)-isogenies.
  $\Rightarrow$ Don't compute unneeded isogenies.
- It can be faster to compute (ℓ,ℓ)-isogenies with $\ell \mid (O_K : \mathbb{Z}[\pi,\overline{\pi}])$ to find new maximal Jacobians when ℓ and $\mathrm{val}_\ell((O_K : \mathbb{Z}[\pi,\overline{\pi}]))$ is small.

“Going up”

- There is p^3 classes of isomorphic curves, but only a very small number ($\#\mathfrak{C}(O_K)$) with $\mathrm{End}(J) = O_K$.
- But there is at most $16p^{3/2}$ isogeny class.

$\Rightarrow$ On average, there is $\approx p^{3/2}$ curves in a given isogeny class.

$\Rightarrow$ If we have a curve in the right isogeny class, try to find isogenies giving a maximal curve!

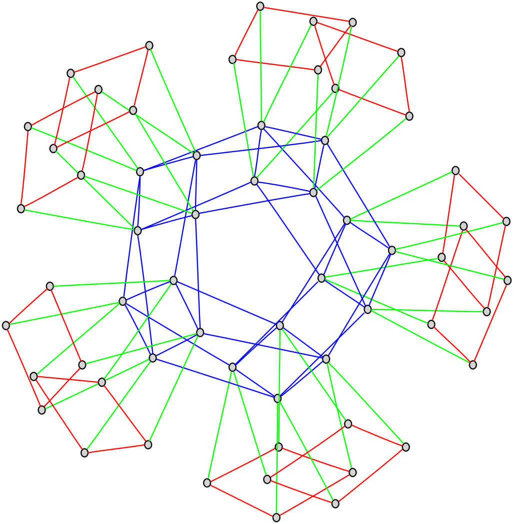
An algorithm for “going up”

1. Let $\gamma \in O_K \setminus \mathrm{End}(J)$. We can assume that $\ell^\infty \gamma \in \mathbb{Z}[\pi, \overline{\pi}]$.
2. Let d be the smallest integer such that $\gamma(J[\ell^d]) \neq \{0\}$, and let $K = \gamma(J[\ell^d])$. By definition, $K \subset J[\ell]$.
3. We compute all (ℓ, ℓ)-isogeneous Jacobians J' where the kernel intersect K. Keep J' if $\#\gamma(J'[\ell^d]) < \#K$ (and be careful to prevent cycles).

- First go up for $\gamma = (\pi^\alpha - 1)/\ell$: this minimize the extensions we have to work with.

Some pesky details

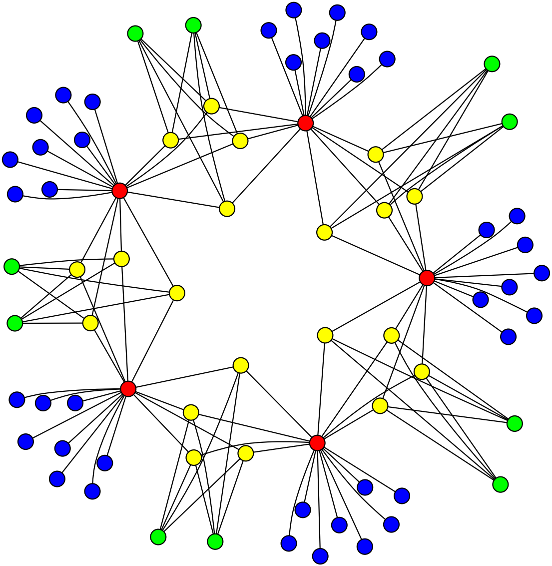
Non maximal cycles ⇒ We try to reduce globally the obstruction for

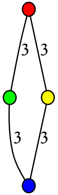


all endomorphisms.

Some pesky details

Local minimums





Some pesky details

- It is not always possible to go up. We would need more general isogenies than (ℓ,ℓ)-isogenies.
- Most frequent case: we can't go up because there is no (ℓ,ℓ)-isogenies at all! (And we can detect this).

The modified CRT algorithm

1. Select a prime p.
2. Select a random Jacobian until it is in the right isogeny class.
3. Go up to find a Jacobian with CM by O_K (if it fails, go back to last step).
4. Use isogenies to find all other Jacobians with CM by O_K.
5. From the invariants of the maximal abelian surfaces, reconstruct $H_i \mod p$.

Sieving the primes

- We throw a prime p for the CRT if detecting if a curve is maximal is too costly, or there is not enough curves where we can “go up”.
- How to estimate this number?
  1. Compute the lattice of orders between $\mathbb{Z}[\pi,\overline{\pi}]$ and O_K. For all such order O such that $(O_K:O)$ is not divisible by any ℓ where there is no (ℓ,ℓ)-isogeny, compute $\mathfrak{C}(O)$.
     This is too costly! (Even computing $\mathrm{Pic}(\mathbb{Z}[\pi,\overline{\pi}])$ is too costly!)
  2. Compute

$$\#\mathfrak{C}(\mathbb{Z}[\pi,\overline{\pi}]) = \frac{c(O_K:Z[\pi,\overline{\pi}])\#\mathrm{Cl}(O_K)\mathrm{Reg}(O_K)(\widehat{O}_K^*:\widehat{\mathbb{Z}}[\pi,\overline{\pi}]^*)}{2\#\mathrm{Cl}(\mathbb{Z}[\pi+\overline{\pi}])\mathrm{Reg}(\mathbb{Z}[\pi+\overline{\pi}])}$$

     and estimate the number of curves as

$$\sum_{d \mid \#\mathfrak{C}(\mathbb{Z}[\pi,\overline{\pi}])} d$$

     (for d not divisible by a ℓ where we can't go up).
- We use a dynamic approach: if a prime discarded earlier is now better than the current prime, go back to this prime.

Exploring the curves

1. Go sequentially through the p^3 Igusa invariants j_1, j_2, j_3. But constructing the curve from the invariants is costly.
2. Construct random curves in Weierstrass form
$$y^2 = a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$
3. If the two torsion is rational (check where $\frac{\pi-1}{2}$ live), construct curves in Rosenhain form
$$y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu).$$
4. If the Hilbert moduli space is rational, construct the j-invariants from the Gundlach invariants (only p^2 invariants, parametrizing the space of curves with real multiplication by K_0).

Finding the denominators

- Use Brunier-Yang formulas to get a multiple of the denominator.
- Do a rationnal reconstruction in K_0^r using LLL.
- Since the Brunier-Yang formula give the denominator for both CM types, both methods are roughly the same.

p	l^d	α_d	# Curves	Estimate	Time (old)	Time (new)
7	2^2	4	7	8	$0.5+0.3$	$0+0.2$
17	2	1	39	32	$4+0.2$	$0+0.1$
23	$2^2,7$	$4,3$	49	51	$9+2.3$	$0+0.2$
71	2^2	4	7	8	$255+0.7$	$5.3+0.2$
97	2	1	39	32	$680+0.3$	$2+0.1$
103	$2^2,17$	$4,16$	119	127	$829+17.6$	$0.5+1$
113	$2^5,7$	$16,6$	1281	877	$1334+28.8$	$0.2+1.3$
151	$2^2,7,17$	$4,3,16$	-	-	0	0
					$3162s$	$13s$

Computing the class polynomial for $K=\mathbb{Q}(i\sqrt{2+\sqrt{2}})$, $\mathfrak{C}(O_K)=\{0\}$.

$$H_1 = X - 1836660096,\quad H_2 = X - 28343520,\quad H_3 = X - 9762768$$

p	l^d	α_d	# Curves	Estimate	Time (old)	Time (new)
29	$3,23$	$2,264$	-	-	-	-
53	$3,43$	$2,924$	-	-	-	-
61	3	2	9	6	$167+0.2$	$0.2+0.5$
79	3^3	18	81	54	$376+8.1$	$0.3+0.9$
107	$3^2,43$	$6,308$	-	-	-	-
113	$3,53$	$1,52$	159	155	$1118+137.2$	$0.8+25$
131	$3^2,53$	$6,52$	477	477	$1872+127.4$	$2.2+44.4$
139	3^5	81	?	486	-	$1+36.7$
157	3^4	27	243	164	$3147+16.5$	-
					$6969s$	$114s$

Computing the class polynomial for $K=\mathbb{Q}(i\sqrt{13+2\sqrt{29}})$, $\mathfrak{C}(O_K)=\{0\}$.

$$H_1 = X - 268435456, \quad H_2 = X + 5242880, \quad H_3 = X + 2015232.$$

p	l^d	α_d	# Curves	Estimate	Time (old)	Time (new)
7	-	-	1	1	0.3	$0+0.1$
23	**13**	84	15	2 (16)	$9+70.7$	$0.4+24.6$
53	7	3	7	7	$105+0.5$	$7.7+0.5$
59	2, **5**	1, 12	322	48 (286)	$164+6.4$	$1.4+0.6$
83	3, 5	4, 24	77	108	$431+9.8$	$2.4+1.1$
103	67	1122	-	-	-	-
107	7, **13**	3, 21	105	8 (107)	$963+69.3$	-
139	$\mathbf{5}^2$, 7	60, 2	259	9 (260)	$2189+62.1$	-
181	3	1	161	135	$5040+3.6$	$4.5+0.2$
197	5, 109	24, 5940	-	-	-	-
199	$\mathbf{5}^2$	60	37	2 (39)	$10440+35.1$	-
223	2, *23*	1, 11	1058	39 (914)	$10440+35.1$	-
227	109	1485	-	-	-	-
233	5, 7, **13**	8, 3, 28	735	55 (770)	$11580+141.6$	$88.3+29.4$
239	7, 109	6, 297	-	-	-	-
257	3, 7, **13**	4, 6, 84	1155	109 (1521)	$17160+382.8$	-
313	3, **13**	1, 14	?	146 (2035)	-	$165+14.7$
373	5, 7	6, 24	?	312	-	$183.4+3.8$
541	2, 7, **13**	1, 3, 14	?	294 (4106)	-	$91+5.5$
571	3, **5**, 7	2, 6, 6	?	1111 (6663)	-	$96.6+3.1$
					56585s	776s

Computing the class polynomial for $K=\mathbb{Q}(i\sqrt{29+2\sqrt{29}})$, $\mathfrak{C}(O_K)=\{0\}$.

$$H_1 = 244140625X - 2614061544410821165056$$

A Dihedral example

- K is the CM field defined by X^4+13X^2+41. $O_{K_0}=\mathbb{Z}[\alpha]$ where α is a root of $X^2-3534X+177505$.
- We first compute the class polynomials over $\mathbb{Z}$ using Spallek's invariants, and obtain the following polynomials in 5956 seconds:

$$H_1 = 64X^2 + 14761305216X - 11157710083200000$$
$$H_2 = 16X^2 + 72590904X - 8609344200000$$
$$H_3 = 16X^2 + 28820286X - 303718531500$$

- Next we compute them over the real subfield and using Streng's invariants. We get in 1401 seconds:

$$H_1 = 256X - 2030994 + 56133\alpha;$$
$$H_2 = 128X + 12637944 - 2224908\alpha;$$
$$H_3 = 65536X - 11920680322632 + 1305660546324\alpha.$$

- Primes used: 59, 139, 241, 269, 131, 409, 541, 271, 359, 599, 661, 761.

Complexity coming from isogenies

Let $\Delta_0 = \Delta_{K_0/\mathbb{Q}}$ and $\Delta_1 = N_{K_0/\mathbb{Q}}(\Delta_{K/K_0}$ so that $\Delta = \Delta_1 \Delta_0^2$.

- The complexity of the going-up step and checking the endomorphism ring is polynomial in the highest prime power dividing the index. For the CRT prime we are using the index is a polynomial in Δ. There is a positive density of prime where the largest prime dividing the index is $O(\Delta^\varepsilon)$ so we can neglect the corresponding cost in the complexity analysis.
- We need horizontal isogenies of small degrees to generate all maximal curves from one. In practice this was always the case (elements of norm polylogarithmic in Δ generates the Shimura class groups).
- At worst, we know that the class group of K^r is generated by totally split primes of norm polylogarithmic in Δ. The typenorm of these elements will yield horizontal isogenies of small degrees.
- The cofactor $\mathfrak{C}/N_\Phi(\mathrm{Cl}(K^r)$ is bounded by $2^{6w(\Delta)+1}$, where $w(\Delta)$ is the number of divisors of Δ. Outside a zero density of very smooth numbers, $w(\Delta) < 2\log\log\Delta$ so we can absorb the factor in the $\widetilde{O}$ notation.

A pessimal view on the complexity of the CRT method in dimension 2

- The degree of the class polynomials is $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$.
- The size of coefficients is bounded by $\widetilde{O}(\Delta_0^{5/2}\Delta_1^{3/2})$ (non optimal). In practice, they are $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$.

$\Rightarrow$ The size of the class polynomials is $\widetilde{O}(\Delta_0\Delta_1)$.

- We need $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ primes, and by Cebotarev the density of primes we can use is $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2}) \Rightarrow$ the largest prime is $p = \widetilde{O}(\Delta_0\Delta_1)$.

$\Rightarrow$ Finding a curve in the right isogeny class will take $\Omega(p^{3/2})$ so the total complexity is $\Omega(\Delta_0^2\Delta_1^2) \Rightarrow$ we can't achieve quasi-linearity even if the going-up step always succeed!

$\Rightarrow$ A solution would be to work over convenient subspaces of the moduli space.

Perspectives

- 6 seconds for 10000 curves is way too slow! Implement this part with `pari`!
- Compute Gundlach invariants for more real quadratic fields.
- In progress: combine the going-up method with Gaetan's sub-exponential endomorphism ring computation. Particularly interesting when a power divides the index.
- More general isogenies than (ℓ,ℓ)-isogenies!