# Improved CRT Algorithm for Class Polynomials in Genus 2

Kristin Lauter, Damien Robert

# IMPROVED CRT ALGORITHM FOR CLASS POLYNOMIALS IN GENUS 2

DAMIEN ROBERT AND KRISTIN E. LAUTER

ABSTRACT. We present a generalization to genus 2 of the probabilistic algorithm in Sutherland [28] for computing Hilbert class polynomials. The improvement over the algorithm presented in [5] for the genus 2 case, is that we do not need to find a curve in the isogeny class with endomorphism ring which is the maximal order: rather we present a probabilistic algorithm for "going up" to a *maximal* curve (a curve with maximal endomorphism ring), once we find *any* curve in the right isogeny class. Then we use the structure of the Shimura class group and the computation of $(\ell, \ell)$-isogenies to compute all isogenous maximal curves from an initial one.

This article is an extended version of the version published at ANTS X.

## 1. INTRODUCTION

Cryptographic solutions to provide privacy and security for sensitive transactions depend on using a mathematical group where the discrete logarithm problem is hard. For example, digital signature schemes or a Diffie-Hellman key exchange may be based on the difficulty of solving the discrete logarithm problem in the group of points on the Jacobian of a genus 2 curve. For this problem to be hard we must ensure that we can choose genus 2 curves over finite fields with an almost prime number of points on the Jacobian of the curve.

One approach to this problem is to construct curves with Jacobian of a given order using the (CM) method of Complex Multiplication. The CM method works by computing invariants of the curve and then reconstructing the curve using the Mestre-Cardona-Quer [23] algorithm. Invariants are computed by constructing their minimal polynomials, called Igusa class polynomials. Computing the invariants is computationally intensive, and there are three known methods for constructing Igusa class polynomials:

(1) the complex analytic method [26, 29, 30, 27];
(2) the Chinese Remainder Theorem method (CRT) [12, 14, 5]; and
(3) the p-adic lifting method [15, 7, 8].

Although the CRT method in genus 2 is currently still by far the slowest of these three methods as measured on small examples which have been computed to date, there is some hope that it may be asymptotically competitive with the other methods based on the history of the evolution of these three methods in genus 1. Asymptotically, and also for space constraint reasons, the (Explicit) CRT method now holds the record in genus 1 for the size of the largest examples computed via that method [28, 13]. In this paper, we propose numerous improvements to the CRT method for computing

2000 *Mathematics Subject Classification.* Primary 11G20; Secondary .

*Key words and phrases.* hyperelliptic curves, Igusa class polynomials, complex multiplication, CRT method, abelian varieties, isogenies.

genus 2 curves, paralleling improvements made by Sutherland [28] to the CRT method in genus 1.

The CRT method works by computing class polynomials modulo many small primes, and then reconstructing the polynomial with rational coefficients (or modulo a much larger prime number) via the Chinese Remainder Theorem (*resp.* the Explicit CRT). The CRT method for computing class polynomials in genus 2 was proposed by [12], with sufficient conditions on the CRT primes to ensure correctness and including an algorithm for computing endomorphism rings for ordinary Jacobians of genus 2 curves which generalized Kohel's algorithm for genus 1 curves. For each small CRT prime $p$, the algorithm loops through all $p^3$ possible triples of Igusa invariants of curves, reconstructing the curve and testing for each curve whether it is in the desired isogeny class and whether its endomorphism ring is maximal. The algorithm for computing endomorphism rings from [12] was replaced by a much more efficient probabilistic algorithm in [14], where a number of examples were given for running times of the computations modulo small CRT primes. [5] introduced the idea of using computable $(3,3)$-isogenies to find other curves in the isogeny class once an initial curve was found, but still searched until finding a curve with maximal endomorphism ring. Another improvement described in [5] was a method to construct other maximal curves using $(3,3)$ isogenies once a first maximal curve is found.

In this paper we present a generalization to genus 2 of the probabilistic Algorithm 1 in Sutherland [28]. The improvement over the algorithm presented in [5] for the genus 2 case, is that, here we do not need to find a curve in the isogeny class with endomorphism ring which is the maximal order: rather we present a probabilistic algorithm for "going up" to a *maximal* curve (a curve with maximal endomorphism ring), once we find *any* curve in the right isogeny class. Then we use the structure of the Shimura class group and the computation of $(\ell,\ell)$-isogenies to compute all isogenous maximal curves from an initial one. Although we cannot prove that the "going up" algorithm succeeds with any fixed probability, it works well in practice, and heuristically it improves the running time of the genus 2 CRT method from $p^3$ per prime $p$ to $p^{\frac{3}{2}}$ per prime $p$.

Let $K$ denote a primitive quartic CM field, and $\Phi$ a CM type. Let $K_\Phi$ denote the reflex CM field, and $K^+$ the totally real subfield of $K$. For a field $K$, let $\mathcal{O}_K$ be the ring of integers of $K$. Let $TN_\Phi$ be the typenorm associated to the CM-type $\Phi$. Informally, the algorithm is as follows, and individual steps will be explained in subsequent sections:

**Algorithm 1.**
*Input: A primitive quartic cyclic CM field $K$ with a CM type $\Phi$, a collection of CRT primes $P_K$ for $K$.*
*Output: Igusa Class Polynomials $H_i(x)$, $i=1,2,3$ in $\mathbb{Q}[x]$ or modulo a prime $q$.*

- *Loop through CRT primes $p \in P_K$:*
  1. *Enumerate hyperelliptic curves $C$ of genus 2 over $\mathbb{F}_p$ until a curve in the right $\mathbb{F}_p$-isogeny class (up to a quadratic twist) is found.*
  2. *Try to go up to a maximal curve from $C$, if it fails go back to Step 1.*
  3. *From a maximal curve $C$, compute all other maximal curves.*
  4. *Reconstruct the class polynomials $H_i(x)$ modulo $p$ from the Igusa invariants of the set of maximal curves.*

- *Recover $H_i(x)$, $i = 1, 2, 3$ in $\mathbb{Q}[x]$ or modulo $q$ using the (Explicit) CRT method once we have $H_i(x)$ modulo $p$ for enough primes $p$.*

For the dihedral case, one new aspect of our algorithm is that we extend to the CRT setting the idea of computing the class polynomials associated to only one fixed CM type $\Phi$ for $K$ [27, Section III.3]. When $K$ is cyclic, this makes no difference, since all isomorphism classes of abelian surfaces with CM by $K$ arise from one CM type; but when $K$ is dihedral, two CM types are needed to find all isomorphism classes of CM abelian surfaces. All three previous versions of the CRT algorithm [12, 14, 5] compute the class polynomials classifying all abelian varieties with CM by $\mathcal{O}_K$ (with either of the two possible CM types in the dihedral case). The advantage of our approach is that it computes only a factor of half the degree of the whole class polynomial. The drawback of this approach is that in the dihedral case, each factor of the class polynomials is defined over $\mathcal{O}_{K_\Phi^+}$ rather than over $\mathbb{Z}$. So once we compute the class polynomials modulo $\mathfrak{p}$ as polynomials in $\mathcal{O}_{K_\Phi^+}/\mathfrak{p}$, the CRT step must be performed in $\mathcal{O}_{K_\Phi^+}$.

A CRT prime $\mathfrak{p} \subset \mathcal{O}_{K_\Phi^+}$ is a prime such that all abelian varieties over $\mathbb{C}$ with CM by $(\mathcal{O}_K, \Phi)$ have good reduction modulo $\mathfrak{p}$. By [25, Section III.13], $\mathfrak{p}$ is a CRT prime for the CM type $\Phi$ if and only if there exists an unramified prime $\mathfrak{q}$ in $\mathcal{O}_{K_\Phi}$ of degree 1 above $\mathfrak{p}$ of principal type norm $(\pi)$ (in particular this implies that $\mathfrak{q}$ is totally split in the class field corresponding to the abelian varieties with CM by $(\mathcal{O}_K, \Phi)$). Moreover, by a theorem of Tate, the isogeny class of these abelian varieties reduced modulo $\mathfrak{p}$ (by [16, Section 3] they have good reduction) is determined by the characteristic polynomial of $\pm\pi$ (here we assume that $\mathcal{O}_K^* = \pm 1$). For efficiency reasons, we will work with CRT primes $\mathfrak{p}$ that are unramified of degree one over $p = \mathfrak{p} \cap \mathbb{Z}$. By [16], the reduction to $\mathbb{F}_p$ of the abelian varieties with CM by $(\mathcal{O}_K, \Phi)$ will then be ordinary. We then make the slight abuse of notation of calling $p$ a CRT prime when there is a CRT prime $\mathfrak{p}$ above it. Note another advantage of restricting to one CM type: to use $p$ for both CM types, $p$ needs to split completely into $p = \mathfrak{p}_1\mathfrak{p}_2$ such that both $\mathfrak{p}_1$ and $\mathfrak{p}_2$ are CRT primes, and there are fewer $p$ which satisfy this stronger requirement.

In addition to the two main contributions of the paper, the "going up" algorithm to find maximal curves, and an improvement to the algorithm to compute maximal curves from maximal curves, we also give improvements to every step of the CRT algorithm. Here we give a brief outline of the paper and a summary of those improvements.

We discuss CRT primes in Section 2, and we explain how to compute the associated isogeny class over $\mathbb{F}_p$ in Section 3.

Step 2 (the "going-up part) of the algorithm is explained in Section 5. We first explain in Section 4 how to compute if a curve is maximal, since this is used in the going-up algorithm. We present some significant improvements over the algorithm from [14]. Step 3 (finding all other maximal curves from one maximal curve) is explained in Section 6. The running times of Steps $1-3$ of the algorithm depend greatly on the CRT prime chosen (for instance the size of the isogeny class can vary). Since we have latitude in the choice of CRT primes, we explain our algorithm for sieving them in Section 7.

As for Step 4, once all maximal abelian varieties with CM by $K$ are found for a given prime $p$, it is easy to compute the associated class polynomials modulo $p$. The

class polynomials depend on the choice of Igusa invariants, and we use the invariants recommended in [27, Appendix 3] which give smaller coefficients than those used in [29, 30, 16]. For the dihedral case the class polynomials must be reconstructed over $\mathcal{O}_{K_\Phi^+}$, and we give more details about this step in Section 8.

Section 9 gives a complexity analysis, and explains how each improvement affects the final complexity. The final complexity bound, while still not quasilinear, is a significant improvement compared to [5]. Finally, examples demonstrating significantly improved running times are given in Section 10.

This paper is the long version of the paper which appeared at ANTS X, and it differs from the short version in the following ways. This paper contains extra sections detailing the definition of a CRT prime (Section 2), new methods for efficiently finding a curve in the right isogeny class (Section 3), and strategies for sieving CRT primes (Section 7), along with providing additional examples in Section 10.

## 2. Conditions on CRT primes

The CRT algorithm computes the Igusa class polynomials modulo primes $p$, for primes satisfying certain conditions: namely that the interpolation polynomials computed from the Igusa invariants of the abelian varieties over $\mathbb{F}_p$ with CM by $\mathcal{O}_K$ are the reductions modulo $p$ of the Igusa class polynomials. As proved in [12], this is satisfied for primes $p$ such that all abelian varieties with CM by $\mathcal{O}_K$ have ordinary reduction and are defined over $\mathbb{F}_p$. We call such a prime $p$ a CRT prime (for the field $K$).

Originally, in [12], the condition given was that CRT primes $p$ should split completely into principal ideals in $K_\Phi$, the reflex field of $K$. This strong condition implied three consequences: since the prime splits completely in $H_{K_\Phi}$, the Hilbert class field of $K_\Phi$, all abelian surfaces with CM by $K$ are defined over $\mathbb{F}_p$ (not just the ones with CM by $\Phi$); since the prime splits completely in $K$, the reduction of the abelian surfaces with CM by $K$ is ordinary and the relative norm equation has a solution.

However, this condition is not always necessary, because the CM points in the moduli space may be defined over $K_\Phi(j_1(A))$, the subfield of $H_{K_\Phi}$ generated by the Igusa invariants of $A$, an abelian surface with CM by $K$.

Moreover, as noted the introduction, in practice we only want to compute the Igusa class polynomials relative to a CM type $\Phi$ (which gives a factor of the full Igusa class polynomials in the dihedral case). For efficiency reasons, we also want the reductions to be ordinary. In this case the class polynomials are defined over $K_\Phi^+$, and we are then looking for CRT primes $\mathfrak{p} \in \mathcal{O}_{K_\Phi^+}$. We call such a prime $\mathfrak{p}$ a CRT prime if all abelian varieties with CM by $(\mathcal{O}_K, \Phi)$ have good ordinary reduction to $\mathcal{O}_{K_\Phi^+}/\mathfrak{p}\mathcal{O}_{K_\Phi^+}$. The results from [16] imply that $p = \mathbb{Z} \cap \mathfrak{p}$ is totally split in $\mathcal{O}_{K_\Phi^+}$. In particular, the interpolation polynomials computed over $\mathcal{O}_{K_\Phi^+}/\mathfrak{p}\mathcal{O}_{K_\Phi^+}$ are actually defined over $\mathbb{F}_p$ and by abuse of notation we call $p$ a CRT prime.

We have the following characterization of CRT primes:

**Theorem 2.** *Let $\Phi$ be a CM type and $p \in \mathbb{Z}$ be a prime number. If there exists an unramified prime $\mathfrak{p}$ in $O_{K_\Phi}$ of degree 1 above $p$ of principal typenorm $(\pi)$ such that $\pi\overline{\pi} = p$, then $p$ is a CRT prime for the CM type $\Phi$.*

*If moreover $p$ splits completely in $\mathcal{O}_{K_\Phi}$ into primes with principal typenorm, then $p$ is a CRT prime for $K$.*

*Proof.* Let $A/\mathbb{C}$ be an abelian variety of CM type $(K, \Phi)$. There exists a model of $A$ defined over the subfield $H_0$ of $H_{K_\Phi}$ generated by the Igusa invariants of $A$. By [25], the condition on $\mathfrak{p}$ implies that it splits completely in this subfield. If we fix one of the primes $\mathfrak{p}_0$ above $\mathfrak{p}$, then by [16], we have that $A_{\mathfrak{p}_0}$ has ordinary reduction defined over $\mathbb{F}_p$. All such abelian varieties with CM by $\Phi$ have ordinary reduction over $\mathbb{F}_p$, so $p$ is a CRT prime.

For the second assertion, we have by the hypothesis that $p$ splits completely as $p = \mathfrak{p}_1\mathfrak{p}_2$ in $\mathcal{O}_{K_\Phi^+}$ (which does not depend of the CM type). By the preceding paragraph, since all primes in $\mathcal{O}_{K_\Phi}$ above $\mathfrak{p}_1$ satisfy the condition of the first part, $p$ is a CRT prime for the CM type $\Phi$. This concludes the cyclic case. Now for the dihedral case, let $\Phi'$ be the other CM type. We will see in Section 2.2, that the reductions of the abelian varieties with CM by $\Phi$ modulo primes in $\mathcal{O}_{K_\Phi}$ above $\mathfrak{p}_2$ correspond exactly to the reductions of the abelian varieties with CM by $\Phi'$ modulo primes in $\mathcal{O}_{K_{\Phi'}}$ above $\mathfrak{p}_1$. So $p$ is also a CRT prime for $\Phi'$. □

2.1. **The cyclic case.** In the cyclic case, there is just one CM type $\Phi$, so the Igusa class polynomials for this CM type are the same as the full Igusa class polynomials for CM by $\mathcal{O}_K$.

A prime $p$ is then a CRT prime if it splits completely in $\mathcal{O}_K$ as $p = \mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3\mathfrak{p}_4$ and $\mathfrak{p}_1$ has a principal typenorm. (Since the other primes are all conjugate to this one, they all also have principal typenorms, and give the same isogeny class.)

2.2. **The dihedral case.** For the dihedral case, a prime number $p$ is a CRT prime if $p$ splits completely as $q_1q_2$ in $K_\Phi^+$, such that $q_1$ splits completely in $K_\Phi$ into primes with principal type norm $(\pi)$ and $\pi\overline{\pi} = p$. A word of warning: when we compute the resulting interpolating Igusa polynomials over $\mathbb{F}_p$, they are the reduction of the Igusa class polynomials modulo $q_1$ and the CRT step will then be computed over $\mathcal{O}_{K_\Phi^+}$, not over $\mathbb{Z}$ as in the cyclic case. We note that if $q_2$ also splits completely into prime ideals with principal type norm, we could use $q_2$ in the CRT rather than $q_1$. We use the sieving algorithms of Section 7 to select the one most practical for the computation.

Now we explain how to adapt the algorithm to compute directly the Igusa class polynomial with CM by $\mathcal{O}_K$ (not just one fixed CM type). In the dihedral case there are two distinct CM types $\Phi_1$ and $\Phi_2$ (conjugate by an element of the Galois group). Let $A_1$ (resp. $A_2$) be an abelian variety with complex multiplication by $\Phi_1$ (resp. $\Phi_2$) and $p$ a prime that splits as $q_1q_2$ in $K_\Phi^+$ (this field does not depend on the choice of the CM type). Note that since the two CM types are conjugate, this gives that $(A_2)_{q_1}$[1] is isogenous to $(A_1)_{q_2}$. Thus both CM types can be handled at once by fixing once and for all the reflex field (corresponding to one fixed CM type $\Phi$) and looking at both primes $q_1$ and $q_2$.

Thus in this case, there are up to two isogeny classes (up to twists) corresponding to maximal curves over $\mathbb{F}_p$, one for each CM type. So to compute the class polynomial corresponding to all curves with CM by $\mathcal{O}_K$, we have to look for CRT primes $p$ that split completely in $K_\Phi^+$ ($p = q_1q_2$) and such that both $q_1$ and $q_2$ split completely, and the primes above them have principal type norm. (The corresponding typenorm of the primes above $q_1$ and $q_2$ will then give the action of Frobenius on both isogeny classes). So that gives fewer CRT primes to work with. Moreover, in Section 7 we

[1]Where by abuse of notation we denote by $(A_2)_{q_1}$ the reduction of $A_2$ modulo a prime above $q_1$ in the moduli field where $A_2$ is defined.

explain how to sieve the CRT primes according to the size of the isogeny class and the ease of endomorphism ring computation and going up. Since we have two isogeny classes here, it is also twice as hard to find good CRT primes $p$. Thus, there are many drawbacks to working over $\mathbb{Z}$ in the dihedral case: we can't distinguish between the two CM types (this would involve distinguishing reductions modulo $q_1$ and $q_2$, which requires working over $\mathcal{O}_{K_\Phi^+}$ where the $q_i$ are defined), so we have to compute the full class polynomials with CM by $\mathcal{O}_K$. This class polynomial has bigger coefficients, there are fewer CRT primes, and it is harder to find good CRT primes among them.

2.3. **Finding the isogeny class.** For a given CRT prime $p$, the algorithm must find all abelian varieties over $\mathbb{F}_p$ with CM by $(\mathcal{O}_K, \Phi)$, i.e. genus 2 curves $H$ such that $\mathrm{End}(\mathrm{Jac}(H)) \simeq \mathcal{O}_K$. These curves lie in several possible isogeny classes, see [12] for the possible number of group orders in the ordinary case. So the algorithm first tries to find a curve in the corresponding isogeny class, (i.e. such that $\mathrm{End}(\mathrm{Jac}(H))\otimes\mathbb{Q} \simeq K$). For that we need a characterization of this isogeny class. By a theorem of Tate, the isogeny class is characterized by the zeta function of a curve in it, or equivalently by the characteristic polynomial of the action of the Frobenius $\pi \in K$. So we need to recover $\pi \in K$ and compute its characteristic polynomial.

If $A_\mathfrak{p}/\mathbb{F}_p$ is the reduction of an abelian variety with complex multiplication by $\Phi$, then [25] shows that the typenorm of $\mathfrak{p} \cap K_\Phi$ gives the action of the Frobenius. This gives the following algorithm:

**Algorithm 3.** *Characterizing the isogeny class.*

**Input:** *An unramified prime $\mathfrak{p}$ (above the prime $p \in \mathbb{Z}$) in $\mathcal{O}_{K_\Phi}$ of degree 1 of principal typenorm.*

**Output:** *The characteristic polynomial of the Frobenius corresponding to the reduction $A_\mathfrak{p}$ of an abelian variety $A$ with CM by $(\mathcal{O}_K, \Phi)$.*

(1) *Compute $TN(\mathfrak{p}) = (\alpha)$.*
(2) *Compute the fundamental unit $\xi$ of $K^+$.*
(3) *Choose an embedding $K \mapsto \mathbb{C}$ and let $|\cdot|$ be the corresponding absolute value.*
(4) *Compute $n = \mathrm{Log}(p/|\alpha|^2)/\mathrm{Log}(|\xi|^2) \in \mathbb{Z}$.*
(5) *Compute $\pi = \alpha\xi^n$.*
(6) *Return the characteristic polynomial of $\pi$.*

**Remark 4.** *The ideal generated by $\alpha$ has relative norm $p$ (by definition of the type norm), but we need to find a generator $\pi$ with complex absolute value equal to $\sqrt{p}$. We adjust $\alpha$ by multiplying by some power of the fundamental unit, $\xi$ in $K^+$. In our algorithm we improve Algorithm 2.1, Step 4, from* [14] *by computing the complex logarithms of $\alpha$ and $\xi$, and then directly adjusting $\alpha$ by the correct power of $\xi$.*

*Note that $-\pi$ is also of norm $\sqrt{p}$, so there are actually two isogeny classes corresponding to Jacobians $J$ with $\mathrm{End}(J) \otimes \mathbb{Q} \simeq K$, but the one corresponding to $-\pi$ is given by a quadratic twist of the first class.*

## 3. Searching for a curve in the isogeny class

In this section we introduce several ways to speed up the computation when searching for a curve in the right isogeny class modulo a given CRT prime. The standard approach in previous implementations has been to loop through all $p^3$ triples of Igusa invariants, generating a genus 2 curve for each triple using Mestre's

algorithm, and then checking whether it is in the right isogeny class by either counting points on the curve to determine the zeta function or by generating random points on the Jacobian and checking that the points are killed by the group order (corresponding to $\pi$ or $-\pi$).

But in our new algorithm, we don't need to loop over all the curves, but only to find a curve in the right isogeny class. For each CRT prime, the probability for a random curve to be in a fixed isogeny class is given by the size of the isogeny class divided by the number of isomorphism classes of curves. We can efficiently sample random curves by taking random monic sextic polynomials with coefficients in $\mathbb{F}_p$. Since there are roughly $p^3$ isomorphism classes of curves and there are $p^6$ sextics, the probability that two randomly chosen sextics define isomorphic curves is $1/p^3$. The advantage to this approach is that randomly generating sextics is a much more efficient way to generate curves than generating the curve equation from triples of Igusa invariants by applying Mestre's algorithm for each triple. This gives the following algorithm:

**Algorithm 5.** *Finding a curve in the isogeny class corresponding to $\pm\pi$.*

*If the characteristic polynomial of $\pi$ is $X^4 + t_0X^3 + t_1X^2 + t_2X + p^2$ then a curve $H$ is in the isogeny class if and only if $\#H(\mathbb{F}_p) = M_1 := p + 1 + t_0$ and $\#\operatorname{Jac}(H) = N_1 := 1 + t_0 + t_1 + t_2 + p^2$. Likewise we define $M_2$ to be the cardinality of a curve in the isogeny class corresponding to the twist $-\pi$ and $N_2$ the cardinality of its Jacobian.*

1. *Take a random hyperelliptic curve $H : y^2 = a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$.*
2. *Test whether $N_iP = 0$ ($i = 1, 2$) for several random elements $P \in \operatorname{Jac}(H)$. (We adapt dynamically the number of random elements $P$ we take for this step depending on the size of $p$.)*
3. *If Step 2 succeeds for all $P$, then test if $\#H = M_i$ and $\#J = N_i$ (for $i = 1$ or for $i = 2$) and return $H$ in case of success. Go back to step 1 otherwise.*

**Remark 6.** *Note though, that because the height of the first Igusa class polynomials is smaller than the height of the second and third class polynomials (with the invariants we are using), it can be computed with fewer CRT primes. Once the first polynomial is known, for subsequent CRT primes it may be better to revert to the method of searching for curves using the Igusa invariants: since the $\deg H_1$ possible values of the first invariant are known, we only need to loop through $p^2 \times \deg H_1$ Igusa triples. This is to be compared with the number of curves we expect to loop through before finding a curve in the right isogeny class.*

Depending on the CRT field $K$, there may be more efficient ways to sample random curves than by taking random monic sextic polynomials.

3.1. **Rosenhain representation.** If the 2-torsion of the Jacobian of any maximal curve in the isogeny class is rational, we can search for curves in Rosenhain form:

**Lemma 7.** *The following assertions are equivalent:*

1. *Every maximal curve can be put in Rosenhain normal form*
$$y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu)$$
*where $\lambda, \mu, \nu \in \mathbb{F}_p$.*
2. *The 2-torsion of a Jacobian of a maximal curve is rational over $\mathbb{F}_p$.*

(3) $\frac{\pi-1}{2} \in O_K$.

*Proof.* It is well known that a hyperelliptic curve of genus 2 can be put into Rosenhain form if and only if the 2-torsion of its Jacobian is rational. So the first assertion implies the second. Now we also have that the 2-torsion on a Jacobian is rational if and only if the Frobenius $\pi$ acts trivially on it, if and only if $\mathrm{Ker}(\pi - 1)$ contains the 2-torsion, if and only if $\frac{\pi-1}{2}$ is an endomorphism. So if one maximal Jacobian $A$ as a rational 2-torsion, then $\frac{\pi-1}{2} \in \mathrm{End}(A) = \mathcal{O}_K$, so the second assertion implies the third. Finally, if $\frac{\pi-1}{2} \in \mathcal{O}_K$, then every maximal Jacobian has $\frac{\pi-1}{2}$ as an endomorphism, so can be put into Rosenhain normal form. □

It is much easier and faster to loop through curves written in Rosenhain form directly. This approach is also better in the sense that it only loops through curves with rational 2-torsion, thus avoiding generating and computing on curves which could not be maximal curves in our isogeny class. Also, a factor of 6 is saved because the ordering of the roots $\lambda, \mu, \nu$ does not matter, and there are 6 ways to permute those 3 roots.

**Remark 8.** *One trade-off we consider in applying the Rosenhain method is the following: while we take random curves over a space one sixth as large as the space of all hyperelliptic curves of genus* 2*, we can only find curves in the isogeny class in Rosenhain form. We could estimate the size of this intersection by checking whether the element* $\frac{\pi-1}{2}$ *is in the suborder* $R \subset \mathcal{O}_K$ *for the various* $R \supset \mathbb{Z}[\pi, \overline{\pi}]$ *and computing the number of curves with endomorphism ring* $R$*, but this is expensive. In practice, we only apply this method when the condition holds for* $R = \mathbb{Z}[\pi, \overline{\pi}]$ *(and hence for all curves in the isogeny class), which is the case for instance when 2 does not divide the index* $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$.

3.2. **Real multiplication.** For a given quartic CM field $K$, write the real quadratic subfield $K^+ = \mathbb{Q}(\sqrt{d})$. For a given CRT prime $p$, any curve over $\mathbb{F}_p$ with CM by $K$ must also have real multiplication by $K^+$. Such curves correspond to CM points associated to $K$ on the Hilbert moduli space associated to $K^+$. Such curves are determined by pairs of Gundlach invariants on the Hilbert moduli space instead of triples of Igusa invariants. Thus the algorithm to search for these curves can be improved by looping through Gundlach invariants instead of Igusa invariants. This approach is more efficient because there are only $p^2$ pairs to search through instead of $p^3$ triples. Formulas for Gundlach invariants and a method for generating genus 2 curves from Gundlach invariants were given in [20]. Those formulas can be used directly in the search step for each CRT prime.

Here the same trade off has to be considered as the Rosenhain method: if $R \bigcap K^+ \not\supset \mathcal{O}_{K^+}$ for various orders $R$ such that $\mathbb{Z}[\pi, \overline{\pi}] \subset R \subset \mathcal{O}_K$, we miss all the curves whose Jacobian have endomorphism ring $R$.

## 4. Checking if the endomorphism ring is maximal

We recall the algorithm described in [12], and describe some improvements. The ideas for computing the endomorphism ring will be used in the going up phase.

4.1. **The algorithm of Eisenträger, Freeman and Lauter.** Let $A/\mathbb{F}_p$ be an ordinary abelian variety of dimension 2 with CM by $K$. Let $O = \mathrm{End}(A)$. We know that $\mathbb{Z}[\pi] \subset \mathbb{Z}[\pi, \overline{\pi}] \subset O \subset O_K$. We want to check if $O = O_K$. First, the Chinese Remainder Theorem gives us the following proposition:

**Proposition 9.** *Let $O$ be an order in $\mathcal{O}_K$. Let $\{1, \alpha_1, \alpha_2, \alpha_3\}$ be a basis of $\mathcal{O}_K$ as a $\mathbb{Z}$-module. Write $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \prod \ell_i^{e_{\ell_i}}$. If $\frac{[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]}{\ell_i^{e\ell_i}} \alpha_j \in O$ for $j = 1, 2, 3$, and all $\ell_i$ dividing the index, then $O = \mathcal{O}_K$.*

We are then reduced to the following problem: for $\gamma \in \mathcal{O}_K$ such that $\ell^e \gamma \in \mathbb{Z}[\pi, \overline{\pi}]$, check if $\gamma \in O$. To simplify the notation,we will often drop the subscript on $e_\ell$ when it is obvious to which $\ell$ we are referring to. We have ([14]):

**Proposition 10.** *Let $O = \mathrm{End}(A)$ and $\gamma \in \mathcal{O}_K$. There exists an integer polynomial $P_\gamma$ such that $\ell^e p\gamma = P_\gamma(\pi)$. Then $\gamma$ is in $O$ if and only if $P_\gamma(\pi) = 0$ on $A[\ell^e]$.*

*Proof.* First note that $[\mathbb{Z}[\pi, \overline{\pi}] : \mathbb{Z}[\pi]] = p$, (see [14]), so that $\ell^e p\gamma \in \mathbb{Z}[\pi]$, which means we can write: $\ell^e p\gamma = P_\gamma(\pi)$. Second, since we are dealing with ordinary abelian varieties over $\mathbb{F}_p$, we have $p \nmid [O_K : \mathbb{Z}[\pi, \overline{\pi}]]$ [14, Proposition 3.7], so that $\gamma \in O \Leftrightarrow p\gamma \in O$. Lastly, by the universal property of isogenies, we have that $P_\gamma(\pi) = 0$ on $A[\ell^e]$ if and only if $p\gamma \in O$ (see [12]). Summing up, we only need to check that $P_\gamma(\pi) = 0$ on $A[\ell^e]$ to check that $\gamma \in O$. □

**Remark 11.** *Since most of the curves in the isogeny class are not maximal, it is more efficient to check $P_\gamma$ on $A[\ell]$, $A[\ell^2]$, ..., rather than directly on $A[\ell^e]$.*

4.2. **Computing the $\ell^e$-torsion.** The main cost of the preceding algorithm is to compute a basis of the $\ell^e$-torsion groups. The cost of such a computation depends on the degree of the extension where the $\ell^e$-torsion points are defined. We have:

**Lemma 12.** *Let $d$ be the degree such that the $\ell$-torsion points of $A$ are defined over $\mathbb{F}_{p^d}$. Then $d \leq \ell^4 - 1$. Furthermore, the $\ell^e$-torsion is all defined over $\mathbb{F}_q$ with $q = p^{d\ell^{e-1}}$.*

*Proof.* Let $\chi_\pi$ be the characteristic polynomial of Frobenius, $\pi$. Then $d$ is the (multiplicative) order of $X$ in the ring $\mathbb{F}_\ell[X]/\chi_\pi(X)$, so $d \leq \ell^4 - 1$. The second assertion follows from [14, Section 6]. □

**Remark 13.** [14, Proposition 6.2] *gives a better bound for maximal abelian surfaces: in that case we have $d \leq \ell^3$, and if $\ell$ is completely split in $\mathcal{O}_K$, we have $d \mid \ell - 1$.*

We will use the following algorithm to compute points uniformly in an $\ell$-primary group containing $A[\ell^e]$:

**Algorithm 14.** *Precomputation:*

*a) Let $d$ be the (multiplicative) order of $X$ in the ring $\mathbb{F}_\ell[X]/\chi_\pi(X)$ and set $d_e = d\ell^{e-1}$.*

*b) Compute $\chi_{\pi^{d_e}}$ as the resultant in $X$ of $\chi_\pi(Y)$ and $Y^{d_e} - X$, and write $\#A(\mathbb{F}_p^{d_e}) = \chi_{\pi^{d_e}}(1) = \ell^e \gamma$ with $\gamma$ prime to $\ell$.*

*Compute uniform random points in the $\ell$-primary component $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$:*

1. *Take a random point $P$ (uniformly) in $A(\mathbb{F}_{p^{d_e}})$;*
2. *return $\gamma P$.*

Algorithm 4.3 of [14] takes random (uniform) points $P$ in $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$ to get random points in $A(\mathbb{F}_{p^{d_e}})[\ell^e]$ by looking at the smallest $k$ such that $\ell^k.P$ is an $\ell^e$-torsion point, then generates enough such random points so that the probability that they generate the full $\ell^e$-torsion is sufficiently high and then tests $P_\gamma$ on these points of $\ell^e$-torsion. The algorithm computes how many points are needed so that

the probability of generating the full $\ell^e$-torsion is greater than $1-\epsilon$ for some $\epsilon > 0$, so the result is not guaranteed (i.e. it is a "Monte-Carlo" algorithm), which is very inconvenient in our setting since we need to test a lot of curves across different CRT primes $p$.

To ensure correctness we can check that the subgroup generated by the points obtained is of cardinality $\ell^{2ge}$, but this is costly. A more efficient way is as follows: $\{P_1, \dots, P_{2g}\}$ is a basis of the $\ell^e$-torsion if and only if $\{\ell^{e-1}P_1, \dots, \ell^{e-1}P_{2g}\}$ is a basis of the $\ell$-torsion. But that can be easily checked by computing the $g(2g-1)$ Weil pairings $e_\ell(\ell^{e-1}P_i, \ell^{e-1}P_j)$ for $i < j$ and testing if we get an invertible matrix. Since Weil pairings can be computed in $O(\log(\ell))$, this is much faster. This is our first improvement, yielding a "Las-Vegas" algorithm.

The second drawback of the approach of [14] is that, although the random points in $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$ are uniform, this is not always the case for the random points in $A(\mathbb{F}_{p^{d_e}})[\ell^e]$. To have a high probability of generating the full $\ell$-torsion then requires taking many random points in $A(\mathbb{F}_{p^{d_e}})[\ell^\infty]$: if $A(\mathbb{F}_{p^{d_e}})[\ell^\infty] = \ell^s$, the algorithm requires $\ell^{s-4e}(-\log(\epsilon))^{1/2}$ random points to succeed with probability greater than $1-\epsilon$. Since generating these points is the most costly part of the algorithm it is best to minimize the number of random points required. Our second improvement is to use an algorithm, implemented in AVIsogenies, due to Couveignes [10] to get uniform random points in $A(\mathbb{F}_{p^{d_e}})[\ell^e]$. Since the full algorithm is described in more detail in [3], we only give an example to illustrate it here.

Suppose that $G$ is an $\ell$-primary group generated by a point $P$ of order $\ell^2$ and a point $Q$ of order $\ell$. Assume that the first random point chosen is $P = R_1$, which gives an $\ell$-torsion point $T_1 = \ell P$. The second random point $R_2$ chosen will be of the form $\alpha P + \beta Q$. In most cases, $\alpha \neq 0$, so the corresponding new $\ell$-torsion point is $T_2 = \alpha \ell P$, a multiple of $T_1$. However we can correct $R_2$ by the corresponding multiple: compute $R_2' = R_2 - \alpha R_1 = \beta Q$. Thus $R_2'$ gives the rest of the $\ell$-torsion except if $\beta = 0$. In our setting we can use the Weil pairing to express a new $\ell$-torsion point in terms of the generating set already constructed (except when we have an isotropic group, in this case we have to compute the $\ell^2$ multiples), and we only need $O(1)$ random points to find a basis. The cost of finding a basis of the $\ell^e$-torsion is then $O(d_e \log p + \ell^2)$ operations if $\mathbb{F}_{p^{d_e}}$.

4.3. **Reducing the degree.** The complexity of finding the basis is closely related to the degree of the extension $d_e$. Let $d_0$ be the minimal integer such that $(\pi^{d_0} - 1) \in \ell\mathcal{O}_K$. Then $d_0 \mid d$, and as remarked in [14], since we only need to check if $O = \mathcal{O}_K$, we can first check that $\frac{\pi^{d_0}-1}{\ell}$ lies in $O$. In other words, we can check that the $\ell$-torsion points of $A$ are defined over $\mathbb{F}_{p^{d_0}}$ rather than over $\mathbb{F}_{p^d}$. If this is the case, the $\ell^e$-torsion points are then defined over an extension of degree $d_0\ell^{e-1}$ of $\mathbb{F}_p$, which allows working with smaller extensions.

Another improvement we implemented to reduce the degree is to use twists. Let $d_0'$ be the minimal integer such that $((-\pi)^{d_0'} - 1) \in \ell\mathcal{O}_K$. Then we have three possibilities $d_0' = d_0$, $d_0' = 2d_0$, $d_0 = 2d_0'$. In the latter case, it is better to replace $A$ by its twist, since the Frobenius of the twist is represented by $-\pi$, and we can compute the points of $\ell^e$-torsion by working over extensions of half the degree.

**Example 15.** *Let* $H : y^2 = 80x^6 + 51x^5 + 49x^4 + 3x^3 + 34x^2 + 40x + 12$ *be a hyperelliptic curve of genus* 2 *defined over* $\mathbb{F}_{139}$ *and* $J$ *the Jacobian of* $H$*. We have* $\mathrm{End}(J) \otimes \mathbb{Q} \cong \mathbb{Q}(i\sqrt{13 + 2\sqrt{29}})$ *and we want to check if* $\mathrm{End}(J)$ *is maximal. In*

*this example, we compute* $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = 3^5$*, so we need to compute* $J[3^5]$*, which lives over an extension of degree* 81*. If we had checked the endomorphism ring of the Jacobian of the twist of* $H$*, we would have needed to work over an extension of degree* 162*.*

4.4. **Reducing the number of endomorphisms to test.** One last improvement to the algorithm of [14] is to use the fact that $\mathrm{End}(A)$ is an order. So if we know that $\gamma \in O$, then we know that the ring $\mathbb{Z}[\pi, \overline{\pi}, \gamma] \subset O$. As an example, if in terms of our basis we have $\alpha_3 = \alpha_2\alpha_1 \mod \mathcal{O}_K^*$, then we only have to check that $\alpha_2$ and $\alpha_1$ are in $O$ (and since our algorithm works locally at $\ell$, we only need that this relation is true locally for that $\ell$ ). We use this idea as follows: suppose that we have checked that $\{\gamma_i, i = \{1, \ldots, k\}\}$ are endomorphisms lying in $O$, and we want to check if $\gamma \in O$. Let $N_1$ be the order of $\gamma$ in $O_K/\mathbb{Z}[\pi, \overline{\pi}, \gamma_i : i = 1, \ldots, k]$, and $N_2$ be the order of $\gamma$ in $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}]$. If we write $N_2 = \prod \ell_i^{e_{\ell_i}}$, we only have to check that $N_2/\ell^{e_\ell}\gamma \in O$ for $\ell \mid N_1$. In fact, if the valuation of $N_1$ at $\ell$ is $e'_\ell$, then we would only need to check that $N_1/\ell^{e'_\ell}\gamma \in O$, which means testing if $N_1\gamma = 0$ on the $\ell^{e'_\ell}$-torsion, where $N_1\gamma$ is a polynomial in $\pi$, $\overline{\pi}$, and the $\gamma_i$ $(i = 1, \ldots, k)$. We write this polynomial as $\frac{N_1}{pN_2}$ times a polynomial in $\pi$, so that we still need to compute the $\ell^{e_\ell}$-torsion.

**Example 16.** *Let* $H : y^2 = 10x^6 + 57x^5 + 18x^4 + 11x^3 + 38x^2 + 12x + 31$ *be a genus* 2 *curve defined 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* $\mathrm{End}(J) = \mathcal{O}_K$*.* $\mathcal{O}_K$ *is generated as a* $\mathbb{Z}$*-module by* $1, \alpha, \beta, \gamma$*, where* $\alpha$ *is of index* 2 *in* $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}]$*,* $\beta$ *of index* 4 *and* $\gamma$ *of index* 40*. The algorithm from* [14] *would check* $J[2^3]$ *and* $J[5]$*. But* $(\mathcal{O}_K)_2 = \mathbb{Z}_2[\pi, \overline{\pi}, \alpha]$*, so we only need to check* $J[2]$ *and* $J[5]$*.*

4.5. **The algorithm.** Incorporating all these improvements yields the following algorithm:

**Algorithm 17.** *Checking that* $\mathrm{End}\, A$ *is maximal.*

**Input:** *An ordinary abelian surface* $A/\mathbb{F}_p$ *with CM by* $K$*.*
**Output:** *The boolean statement:* $\mathrm{End}\, A = \mathcal{O}_K$*.*

1. *Choose a basis* $\{1, \alpha_1, \alpha_2, \alpha_3\}$ *of* $\mathcal{O}_K$ *and a basis* $\{1, \beta_1, \beta_2, \beta_3\}$ *of* $\mathbb{Z}[\pi]$ *such that* $\beta_1 = c_1\alpha_1$*,* $\beta_2 = c_2\alpha_2$*,* $\beta_3 = c_3\alpha_3$ *and* $c_1, c_2, c_3 \in \mathbb{Z}$ *with* $c_1 \mid c_2 \mid c_3$*.*
2. *(Checking where the* $\ell$*-torsion lives.) For each* $\ell \mid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ *do*
   (a) *Let* $d_\ell$ *be the smallest integer such that* $\pi^{d_\ell} - 1 \in \ell\mathcal{O}_K$*, and* $d'_\ell$ *be the smallest integer such that* $(-\pi)^{d'_\ell} - 1 \in \ell\mathcal{O}_K$*. If* $d'_\ell < d_\ell$*, switch to the quadratic twist.*
   (b) *Compute a basis of* $A[\ell](\mathbb{F}_{p^{d_\ell}})$ *using the algorithm from* [3]*.*
   (c) *If this basis if of cardinality (strictly) less than four, return false.*
   (d) *(Checking the generators of* $\mathcal{O}_K$*.) For* $i = 1, 2, 3$ *do*
      (i) *Let* $N_1$ *be the order of* $\alpha_i$ *in* $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}, \alpha_j : j < i]$ *and* $N_2$ *the order of* $\alpha_i$ *in* $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}]$*.*
      (ii) *If* $\ell \mid N_1$*, let* $e_i$ *be the* $\ell$*-valuation of* $N_2$ *and write* $pN_2\alpha_i$ *as a polynomial* $P_i(\pi)$*.*
      (iii) *Compute a basis of* $A(\mathbb{F}_{p^{d_\ell \ell^{e_i - 1}}})[\ell^{e_i}]$*.*
      (iv) *If* $P_i(\pi) \neq 0$ *on this basis, return false.*
3. *Return true.*

**Remark 18.** *Comments on the algorithm:*

- *With the way we choose the basis of $\mathcal{O}_K$, we have $e_1 \leq e_2 \leq e_3$ (for each $\ell$ dividing the index), so that when we abort early, we may not have the full $A[\ell^{e_3}]$-torsion to compute. Likewise, rather than going through increasing $\ell$, we could go through increasing degrees.*
- *Since we will apply this algorithm to a lot of different abelian varieties, we can precompute everything that is only related to $\mathcal{O}_K$ and $\mathbb{Z}[\pi, \overline{\pi}]$. Then for each abelian variety $A/\mathbb{F}_p$ we want to test, we only have to compute for all $\ell \mid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ the subgroups $A[\ell](\mathbb{F}_{p^{d_\ell}})$ and then subgroups $A(\mathbb{F}_{p^{d_\ell \ell^{e_i - 1}}})[\ell^{e_i}]$ for $i = 1, 2, 3$, testing polynomials of the Frobenius on them. Moreover, if $\ell$ is such that $\ell^2 \not| [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ then $(\mathcal{O}_K)_\ell / \mathbb{Z}_\ell[\pi, \overline{\pi}]$ is cyclic, so that if $\pi^{d_\ell} - 1 \notin \ell\mathbb{Z}[\pi, \overline{\pi}]$, we only need to check that the $\ell$-torsion is defined over $\mathbb{F}_{p^{d_\ell}}$ (see [14]).*

4.6. **Complexity.** The complexity is measured in terms of operations in the base field, $\mathbb{F}_p$, neglecting logarithmic factors $\log(p)$. Since the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ is bounded by a polynomial in $p$ by [14, Proposition 6.2], evaluating the polynomials $P_i(\pi)$ (of degrees at most 3) is done in logarithmic time. The most expensive part of the algorithm is then the computation of $A[\ell^e]$, for the various $\ell$ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ where $e$ is at most the $\ell$-valuation of the index. According to Lemma 12 and Remark 13, the $\ell^e$-torsion points live in an extension of degree at most $d = \ell^{e+3}$. Since $\#A(\mathbb{F}_{p^d}) = p^{2d(1+\epsilon)}$, computing a random point in $A(\mathbb{F}_{p^d})[\ell^e]$ takes $\tilde{O}(d^2)$ operations in $\mathbb{F}_p$. Correcting this random point requires some pairing computations, and costs at most $O(\ell^2)$ (in case the first points give an isotropic group). Since we need $O(1)$ such random points, the global cost is given by the following proposition (we will only need a very rough bound for the complexity analysis in Section 9):

**Proposition 19.** *Let $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \prod \ell_i^{e_i}$ be the decomposition of the index into prime factors. Then checking if an abelian surface in the isogeny class is maximal can be done in time $\sum \tilde{O}(\ell_i^{2e_i+6})$.*

**Remark 20.** *One can compare to* [14, Proposition 4.6] *to see the speedup we gain in the endomorphism ring computation. We note that our method is exponential in the discriminant, while in* [4] *one can find a subexponential algorithm to compute the endomorphism ring of an ordinary abelian surface. In ongoing work with Gaetan Bisson, we have developed a method that combines the going up algorithm of the next section with his endomorphism ring algorithm. Since we still need to take $\ell$-isogenies for $\ell \mid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ in the going up step, this approach is mainly interesting when the index is divisible by a power of a prime.*

## 5. Going up

"Going up" is the process of finding genus 2 curves with maximal endomorphism ring by moving from *any* curve in the isogeny class via isogenies to a maximal one. This is not always possible and we will explain some obstructions. One difficulty was already illustrated in [5, Example 8.3], where it was shown that there can be cycles in the isogeny graph involving some non-maximal curves. Clearly, when trying to "go up", the algorithm should avoid making cycles in the graph, and we propose one method to avoid that. Further difficulties arise from the fact that the graph of rational $(\ell, \ell)$ isogenies can be disconnected and there can be some nodes with no rational $(\ell, \ell)$-isogenies. This is an important caveat, as this means that our method

for "going-up" will not always succeed, so we only have a probabilistic algorithm, and we cannot currently estimate the probability of failure.

As noted in [14], for the type of fields we can deal with via the CRT method, the cost of going through $p^3$ Jacobians is dominant compared to checking if the endomorphism ring is maximal (this imbalance is magnified in our case due to our faster algorithm to compute the $\ell^e$-torsion). In our algorithm, we try to find a random curve in the isogeny class, and we try to select $p$ so that the probability of finding a curve in the right isogeny class is of magnitude $p^{3/2}$. In practice, finding one such curve is still the dominant aspect, which explains why we can afford to spend a lot of effort on "going up" from this curve.

The algorithm we propose for "going up" is made possible by the techniques developed in [22, 9, 24] for computing rational $(\ell,\ell)$-isogenies between abelian varieties over finite fields. If $A$ is an ordinary abelian variety with CM by $K$, then for each $\ell$ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$, we try to find an $(\ell,\ell)$-isogeny path starting from $A$ and going to $A'$ such that $(\mathcal{O}_K)_\ell = \mathrm{End}(A')_\ell$. If this is possible, we let $A = A'$ in the next step (going to the next $\ell$). A rather inefficient method for finding $A'$ would be to use the algorithm for computing endomorphism rings which was detailed in the preceding section (modified to handle the case of non maximal orders), compute the endomorphism ring of $\mathrm{End}(A)$ and the $(\ell,\ell)$-isogenous varieties $A'$, and keep $A'$ if its endomorphism ring is bigger than the one of $A$. In this section we will describe a more efficient algorithm, that combines the endomorphism ring checks of the preceding section with a going up phase. Since we are working locally in $\ell$, we may as well suppose that we are working over $\mathbb{Z}_\ell$.

5.1. **Going up for one endomorphism.** In this section, we suppose that we have an element $\alpha' \in \mathcal{O}_K$ such that $\gamma\ell^e\alpha' \in \mathbb{Z}[\pi]$ with $\gamma$ prime to $\ell$. Starting from an abelian variety $A$ in the isogeny class, we want to find an abelian variety $A'$ such that $\frac{\alpha}{\ell^e} \in \mathrm{End}(A')$ (or equivalently that $\alpha' \in \mathrm{End}(A')$ locally at $\ell$).

We saw in Section 4 that $\frac{\alpha}{\ell^e}$ is in the endomorphism ring of $A$ iff $\alpha(A[\ell^e]) = 0$, and we know how to compute this subgroup. More generally, we let $N = \#\alpha(A[\ell^e])$. We think of $N$ as a way to measure the "obstruction" to "$\frac{\alpha}{\ell^e} \in \mathrm{End}(A)$". Our algorithm is as follows: for each $(\ell,\ell)$-isogenous $A'$, if $N' = \#\alpha(A'[\ell^e])$, then we replace $A$ by $A'$ if $N' < N$. We iterate this process until $N = 1$, in which case we have succeeded, or until we are stuck, in which case we try to find a new random abelian variety in the right isogeny class.

Rather than computing directly the obstruction $N = \#\alpha(A[\ell^e])$, we can compute the partial obstructions $N^\epsilon = \#\alpha(A[\ell^\epsilon])$ for $\epsilon \leq e$. Starting from $\epsilon = 1$, we take isogenies until we find an abelian variety $A$ with $N^\epsilon = 1$, which means that $\frac{\alpha}{\ell^\epsilon} \in \mathrm{End}(A)$. We will now try to take isogenies to reduce the obstruction of higher degree $N^{\epsilon+1}$. Let $k = \alpha(A[\ell^{\epsilon+1}]) \subset A[\ell]$. The following lemma helps us select the isogeny we are looking for:

**Lemma 21.** *Let $A'$ be an abelian variety isogenous to $A$, such that $\#\alpha(A'[\ell^{\epsilon+1}]) < \#\alpha(A[\ell^{\epsilon+1}])$. Then the kernel of the isogeny $A \to A'$ intersects non trivially with $k = \alpha(A[\ell^{\epsilon+1}])$.*

*Proof.* Let $f : A \to A'$ be a rational isogeny between $A$ and $A'$. Then since $\alpha$ is a polynomial in the Frobenius, we have $\alpha \circ f = f \circ \alpha$. In particular, $f$ maps $\alpha(A[\ell^{\epsilon+1}])$ to $\alpha(A'[\ell^{\epsilon+1}])$. If $\#\alpha(A'[\ell^{\epsilon+1}]) < \#\alpha(A[\ell^{\epsilon+1}])$ then there exists $x \in \mathrm{Ker}\, f \cap \alpha(A[\ell^{\epsilon+1}])$. □

This gives the following algorithm:

**Algorithm 22.** *Going up for one endomorphism* $\frac{\alpha}{\ell^\epsilon}$.

**Input:** *An ordinary abelian variety* $A/\mathbb{F}_p$ *with CM by* $K$.
**Output:** *An abelian variety* $A'/\mathbb{F}_p$ *such that* $\frac{\alpha}{\ell^\epsilon} \in \mathrm{End}\, A$ *or* fail.

(1) *Set* $\epsilon = 1$.
(2) *Compute* $N^\epsilon = \#\alpha(A[\ell^\epsilon])$.
(3) *If* $N = 1$, *then if* $\epsilon = e$ *return* $A$. *Otherwise (*$\epsilon < e$*) set* $\epsilon := \epsilon + 1$, *and go back to Step 2.*
(4) *Else (*$N > 1$*) let* $\mathcal{L}$ *be the list of all rational maximal isotropic subgroups of* $A[\ell]$ *which intersect non trivially with* $\alpha(A[\ell^\epsilon])$. *For* $k \in \mathcal{L}$ *do*
   (a) *Compute* $A' = A/k$. *Let* $N' = \#\alpha(A'[\ell^\epsilon])$. *If* $N' < N$, *set* $A = A'$ *and go back to Step 2.*
(5) *(We are stuck) Return* fail.

**Remark 23.** *As in Section 4 we let* $d_0$ *be the minimal integer such that* $(\pi^{d_0} - 1) \in \ell\mathcal{O}_K$ *and* $d$ *the minimal integer such that* $(\pi^d - 1) \in \ell\mathbb{Z}[\pi]$. *Then the* $\ell^\epsilon$*-torsion points of* $A$ *are defined over an extension of degree* $d\ell^{\epsilon-1}$. *If moreover* $\frac{\pi^{d_0}-1}{\ell} \in \mathrm{End}(A)$ *they are actually defined over an extension of degree* $d_0\ell^{\epsilon-1}$.

*Therefore when we try to go up globally for all endomorphisms* $\alpha$, *the first step is to try to go up for the endomorphism* $\frac{\pi^{d_0}-1}{\ell}$. *During the algorithm, the obstruction* $N$ *is given by the size of the kernel of* $\pi^{d_0} - 1$, *whose rank is* $2g$ *minus the rank of the* $\ell$*-torsion points defined over* $\mathbb{F}_{p^{d_0}}$. *So we compute the size of a basis of* $A[\ell](\mathbb{F}_{p^{d_0}})$ *and take isogenies, where this size increases until we find the full rank.*

5.2. **Going up globally.** Let $\{1, \frac{\alpha_i}{\ell^{e_i}}\}$ $(i = 1, 2, 3)$ be generators for the maximal order $(\mathcal{O}_K)_\ell$ over the subring $\mathbb{Z}_\ell[\pi, \overline{\pi}]$, where $\alpha_i \in \mathbb{Z}_\ell[\pi, \overline{\pi}]$. Starting from an abelian variety $A$ in the isogeny class, we want to find an abelian variety which is maximal at $\ell$.

We could apply Algorithm 22 for each $\frac{\alpha_i}{\ell^{e_i}}$, but it does not guarantee that the endomorphisms already defined on $A$ stay defined during the process, so we would observe loops on non maximal abelian varieties with this method. Moreover we want to reuse the computations of $A[\ell^\epsilon]$ which are the expensive part of the process.

If $N_i = \#\alpha_i(A[\ell^{d_i}])$ for $i = 1, 2, 3$ is the obstruction corresponding to $\alpha_i$, we define $N$ to be the global obstruction $N = \sum N_i$. We can then adapt the same method: for each $(\ell, \ell)$-isogenous $A'$, if $N'_i = \#\alpha_i(A'[\ell^{d_i}])$, then we replace $A$ by $A'$ if $\sum N'_i < \sum N_i$. We iterate this process until all the $N_i = 1$, in which case we go to the next $\ell$, or until we are stuck, in which case we try to find a new random abelian variety in the right isogeny class.

As before, if $e = \max(e_1, e_2, e_3)$ we first compute $A[\ell^\epsilon]$ and the partial obstructions $N_i^\epsilon = \#A[\ell^{\min(\epsilon, e_i)}]$ $(i = 1, 2, 3)$. We do the same for the $(\ell, \ell)$-isogenous abelian varieties, and switch to the new one if $\sum N_i^\epsilon$ decreases (strictly). This allows working with smaller torsion in the beginning steps.

The level, $\epsilon$, of the individual obstruction we are working on depends on the endomorphism considered, so if we get stuck on level $\epsilon$, we may have to look at level $\epsilon + 1$ even if not all endomorphisms $\frac{\alpha_i}{\ell^\epsilon}$ are defined yet. For instance (in this example we suppose we only deal with two generators) there are cases where $N_1^\epsilon = 1$, $N_2^\epsilon \neq 1$ and $N'^\epsilon_1 = 1$, $N'^\epsilon_2 = N_2^\epsilon$ for all $(\ell, \ell)$-isogenous abelian varieties $A'$, so we are stuck on level $\epsilon$. However we can still find an isogenous $A'$ such that $N'^{\epsilon+1}_1 < N_1^{\epsilon+1}$.

Finally, as in Remark 23, we first try to go up in a way that increases the size of $A(\mathbb{F}_{p^{d_0}})[\ell]$. If we are unlucky and get stuck, we switch to the computation of the full $\ell$-torsion over $\overline{\mathbb{F}}_p$. This method allows working over the smallest extension to compute $A[\ell^e]$ as soon as possible.

A summary of the algorithm with the notation from above is given below:

**Algorithm 24.** *Going up.*

**Input:** *An ordinary abelian surface $A/\mathbb{F}_p$ with CM by $K$.*
**Output:** *An abelian variety $A'/\mathbb{F}_p$ with* $\operatorname{End} A = \mathcal{O}_K$ *(locally at $\ell$) or* fail.

(1) *(Special case for the endomorphism $\frac{\pi^{d_0}-1}{\ell}$) Compute a basis $B$ of $A(\mathbb{F}_{p^{d_0}})[\ell]$. If $\#B < 2g$, compute a basis $B'$ of $A'(\mathbb{F}_{p^{d_0}})[\ell]$ for each $(\ell,\ell)$-isogenous abelian variety $A'$. If $\#B' > \#B$, restart the algorithm with $A' = A$. If $\#B = 4$ or we get stuck, go to the next step.*
(2) *Set $\epsilon = 1$.*
(3) *Compute*[2] *$N_i^\epsilon = \#\alpha_i(A[\ell^{\min(\epsilon,e_i)}])$ for $i = 1,2,3$.*
(4) *If $\{N_i : i = 1,2,3\} = \{1\}$ then if $\epsilon = \max(e_i : i = 1,2,3)$ return $A$. Else set $\epsilon := \epsilon + 1$ and go back to Step 3.*
(5) *Else let $\mathcal{L}$ be the list of all rational maximal isotropic kernels of $A[\ell]$ which intersect non trivially with one of the $\alpha_i(A[\ell^{\min(\epsilon,e_i)}])$. For $k \in \mathcal{L}$ do*
   (a) *Compute $A' = A/k$. Let $N_i' = \#\alpha_i(A'[\ell^{\min(\epsilon,e_i)}])$. If $\sum N_i' < \sum N_i$, restart the algorithm with $A = A'$ (but do not reinitialise $\epsilon$ in Step 2).*
(6) *If we get stuck and $\epsilon < \max(e_i : i = 1,2,3)$, set $\epsilon := \epsilon + 1$ and go back to Step 3. Otherwise return* fail.

5.3. **Cost of the "going-up" step.** We will see in the examples that the "going-up" step is a very important part in speeding up the CRT algorithm in practical computations. However, since it is doomed to fail in some cases (see Remark 26), we need to check that it will not dominate the complexity of the rest of the algorithm, (so that in theory there is no drawback in using it), and thus we need to estimate the cost of the "going-up" step.

The going up phase is a mix of endomorphism testing and isogeny computations. We already analysed the cost of the endomorphism testing in the preceeding section. For the isogeny computation, the points in the kernel of rational $(\ell,\ell)$-isogenies live in an extension of degree at most $\ell^2 - 1$. Transposing the analysis of Section 4.6 to this case shows that the computation of all points in these kernels takes at most $\tilde{O}(\ell^4)$. There are at most $O(\ell^3)$ such kernels, and each isogeny computation takes at most $\tilde{O}(\ell^4)$ operations in the extension. The final cost is at most $\tilde{O}(\ell^9)$ for computing all isogenies. For each of the $O(\ell^3)$ isogenous abelian varieties we do (part of) the endomorphism ring computation, which is $\ell^{2e+6}$ according to Section 4.6. Since the global obstruction computed is of size $O(\ell^e)$, we do at most $O(e)$ steps. The global complexity is then:

**Proposition 25.** *Let $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]] = \prod \ell_i^{e_i}$ be the decomposition of the index into prime factors. Then the going up phase either fails or is done in at most $\tilde{O}(\sum \ell_i^{2e_i+9})$ operations in the base field.*

**Remark 26.** *It is important to note that the going up phase does not always succeed. We will give some examples of that in Section 10. First, as noted in the introduction*

[2]The degree of the extension where the full $\ell^\epsilon$-torsion is defined depends on whether Step 1 succeeded.

*of the section, the $(\ell,\ell)$-isogeny graph is not always connected, so if we start with a curve not in the same component of a maximal curve, there is no way to find the maximal curves using only $(\ell,\ell)$-isogenies. Second, even if the curve is in the same component, finding a maximal curve may involve going through isogenous curves that increment the global obstruction, so the going up algorithm would not find it.*

*In practical computations we observed the following behavior: in the very large majority of the cases where we were not able to go up, there actually did not exist any rational $(\ell,\ell)$-isogenies for any curve in the isogeny class. If $\chi_\pi$ is the characteristic polynomial, this can be detected by the fact that $\chi_\pi$ does not factor modulo $\ell$ as $\chi_\pi = P\overline{P} \pmod{\ell}$ (where $\overline{P}$ is the conjugate of $P$ under the action $\pi \to p/\pi$, which sends the Frobenius to the Verschiebung). In this situation, there is no way to go up even locally at $\ell$. This gives a criterion to estimate whether one can go up for this $\ell$.*

## 6. Computing maximal curves from maximal curves

Once a maximal curve in the isogeny class has been found via the random search and "going up" steps, we use isogenies to find the other maximal curves. The set of maximal curves in the isogeny class corresponding to a fixed CM type $\Phi$ is a principal homogeneous space under the action of the Shimura class group

$$\mathfrak{C} = \{(I,\rho) \mid I \text{ a fractional } \mathcal{O}_K\text{-ideal with } I\overline{I} = \rho,\ \rho \in K^+ \text{ totally positive}\}/K^*,$$

associated to the primitive quartic CM field $K$, which acts by isogenies (see for instance [5, Section 3]).

However, using AVIsogenies we can only compute isogenies with a maximal isotropic kernel. In terms of the Shimura class group, the lemma below shows that this means that we can only compute the action corresponding to (equivalences classes) of elements of the form $(I,\ell)$ where $I$ is an ideal in $K$ and $\ell$ a prime number.

**Lemma 27.** *Let $(I,\rho)$ be an element of the Shimura class group $\mathfrak{C}$ and $\ell$ a prime. Then the action of $(I,\rho)$ on a maximal abelian variety $A$ corresponds to an isogeny with maximal isotropic kernel in $A[\ell]$ if and only if $\rho = \ell$ (so if and only if $I$ has relative norm $\ell$).*

*Proof.* This follows from the construction of the action of $\mathfrak{C}$ on the set of maximal abelian varieties. The action is given by the isogeny $f : \mathbb{C}^2/\Lambda \to \mathbb{C}^2/I\Lambda$ and moreover the action of $\overline{I}$ corresponds to the dual isogeny $\widehat{f}$ (here we identify the abelian variety $A$ with its dual $\widehat{A}$ via the principal polarization induced from the CM data). Since $\ell$ is prime, the isogeny corresponding to $I$ is an $(\ell,\ell)$ isogeny if and only if $I\overline{I} = \rho = \ell$. □

Therefore to ensure that we can find all other maximal curves using this type of isogeny we make the following heuristic assumption. Here $\Delta$ is the discriminant of $K$:

**Assumption.** *The Shimura class group $\mathfrak{C}$ is generated by elements of the form $(I,\ell)$ where $\ell$ is a polynomial in $\log\Delta$.*

**Justification:** We have tested this assumption on numerous examples. The assumption on the size of the isogenies will be used in the complexity analysis. At worst, we know (under GRH) that the class group of the reflex field is generated by prime ideals of degree one and of norm polynomial in $\log\Delta$ [1, Theorem 1]. (Note that the discriminant of the reflex field is $O(\Delta^2)$). But if $I$ is such an ideal of $\mathcal{O}_{K_\Phi}$

of norm prime to $p$, then the element $(TN(I), N(I))$ will give a horizontal isogeny. So we will at least be able to compute all the maximal curves that are deduced from the first one by an action coming from the type norm. As we will see in the complexity analysis in Section 9, this is sufficient for most discriminants $D$.

**Lemma 28.** *Let $A$ be an ordinary abelian surface with $\mathrm{End}(A) \otimes \mathbb{Q} = K$, and let $f : A \to B$ be an isogeny of degree prime to $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. Then $\mathrm{End}(A) = \mathrm{End}(B)$.*

*Proof.* Let $d$ be the smallest integer that factorizes through $f$, so $d = f\tilde{f}$. By assumption $d$ is prime to the index. If $\alpha \in \mathrm{End}(A)$, then $f \circ \alpha \circ \tilde{f} = d\alpha$ is an endomorphism of $B$. Since $[\mathcal{O}_K : \mathrm{End}(B)]$ is prime to $d$, we have that $\alpha \in \mathrm{End}(B)$. The same argument shows that $\mathrm{End}(B) \subset \mathrm{End}(A)$, so $\mathrm{End}(A) = \mathrm{End}(B)$. □

Note that we can precompute generators of the Shimura class group since this data does not depend on the current prime $p$. We want to find generators of relative norm a prime $\ell \in \mathbb{Z}$ with $\ell$ as small as possible since this will directly influence the time spent to find the other maximal curves.

Now for a CRT prime $p$, there may exist among the generators we have chosen some that divide the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. We can either find other generators (whose norm will be bigger), or still try to use the precomputed generators. In this case, if such a generator has norm $\ell$, then not all new $(\ell, \ell)$-isogenous abelian varieties will be maximal, so we have to use the algorithm of Section 4 to test which of them is maximal. In that case, after the isogeny is applied, the $\ell^e$-torsion must again be computed (with the notation of Section 5), along with the action of the generators of $(\mathcal{O}_K)_\ell$ over $\mathbb{Z}[\pi, \overline{\pi}]_\ell$. The trade-off depends then on the degree of the extension field required to compute the $\ell^e$-torsion for small $\ell$ dividing the index versus the degree of the field of definition for the points in the kernel of the $\ell$-isogeny for $\ell$ not dividing the index.

Finally, we can also use the group structure of the Shimura class group as follows: suppose that we have computed maximal curves corresponding to the action of $\alpha_1, \ldots, \alpha_t \in \mathfrak{C}$, and we want to find new maximal curves by computing $(\ell, \ell)$-isogeny graphs starting from these curves. Then if $\mathfrak{C}(\ell)$ is the set of elements of the form $(I, \ell)$ in $\mathfrak{C}$, then the number of maximal curves that we can find in this way is the cardinality of the subgroup generated by the $\alpha_i$ and $\mathfrak{C}(\ell)$. In particular, as soon as we reach this number, we can stop the computation since it will not yield any new maximal curves. This is particularly useful when $\ell$ divides the index since it avoids some endomorphism tests. In the isogeny graph computation done by AVIsogenies, each node is computed twice since there are two edges between adjacent nodes (corresponding to the isogeny and the dual). Here since we know the number of nodes, we can abort the computation early.

We thus obtain the following algorithm:

**Algorithm 29.** *Finding all maximal curves from one maximal curve.*

**Input:** *An ordinary abelian variety $A/\mathbb{F}_p$ with CM by $(\mathcal{O}_K, \Phi)$.*
**Output:** *All abelian varieties over $\mathbb{F}_p$ with CM by $(\mathcal{O}_K, \Phi)$.*

***Precompute*** *a set of generators of the Shimura class group with relative norm $\ell$ as small as possible (the set is not chosen to be minimal, on the contrary we want some redundancy). For each of the generators, compute the extension degree of the field of definition of the geometric points of the kernel corresponding to this generator.*

(1) *For each generator of (relative) norm $\ell$ dividing the index, replace the previous degree by the degree of the extension where the $\ell^e$-torsion lives (usually $e$ is the $\ell$-valuation of the index, but the tricks from Section 4 can sometimes reduce it).*
(2) *Sort the generators according to the corresponding degrees to get a list $(g_1, \dots, g_n)$.*
(3) *For each generator $g_i$ on the list, let $\ell_i$ be its norm and do*
   (a) *Compute the varieties $(\ell_i, \ell_i)$-isogenous to the one already found. If $\ell_i$ divides the index, then do an endomorphism ring computation from Section 4 and keep only the maximal curves*
   (b) *Repeat until the number of (maximal) abelian varieties is $\#\langle \mathfrak{C}(\ell_1), \dots, \mathfrak{C}(\ell_i)\rangle$.*

## 7. Strategies for sieving CRT primes p

Since we have some latitude in the CRT primes $p$, we can sieve the prime to use. For instance, we will reject a prime $p$ if the size of the isogeny class is too small, or if computing the endomorphism ring or going up is too costly for this prime. We use a dynamic approach: we reevaluate each discarded CRT prime against the new ones found. In this section, we explain how we estimate the difficulty of the computation associated to one CRT prime.

7.1. **Cost of testing if a curve is maximal.** Before using a prime for the CRT computation, we first need to check if testing if a curve is maximal is too expensive. For this, we compute which subgroup $A[\ell^e]$ we would have to compute to test if $A$ is maximal as in Section 4, and on which extension the points of these subgroups live. As already remarked, for $\ell$ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, $e$ is usually the valuation of $\ell$ in the index, but some of the tricks from Section 4 can reduce it.

If the extension on which we need to do the computation is too large, we exclude the prime $p$. We will explain later how we estimate whether computing the endomorphism ring is too large compared to the current parameters.

7.2. **Size of the isogeny class.** For each CRT prime $p$, the first phase of our algorithm relies on first finding a genus 2 curve over $\mathbb{F}_p$ in the right isogeny class. The larger the isogeny class, the larger the probability of finding a curve in the right isogeny class quickly. There are $p^3$ isomorphism classes of genus 2 curves over $\mathbb{F}_p$, and since the area of Figure 10.1 in [21] is $32/3$, there are approximately $(32/3)p^{3/2}$ isogeny classes. We could then expect that on average, each isogeny class has roughly $\frac{3p^{3/2}}{32}$ curves.

However it happens that, for a fixed primitive CM field $K$, for some primes $p$ the isogeny class corresponding to the Frobenius element $\pi$ can be unfortunately small. In those cases, our algorithm has a lower chance of finding a curve in the isogeny class quickly, so it is most likely more efficient for the algorithm to skip that CRT prime and proceed to another prime where the chance of finding a curve in the right isogeny class is bigger.

To determine whether a potential CRT prime should be skipped or not, we need to estimate the size of the isogeny class. If the estimated size is not at least a certain fraction of $p^{3/2}$ then we skip the prime. The size of the isogeny class is given as $\sum_O \#\mathfrak{C}(O)$, the sum of the size of the Shimura class group associated to each order between $\mathbb{Z}[\pi, \overline{\pi}]$ (stable by the complex conjugation). Of course we don't want to compute the Shimura class group (even computing only the class group can be too

expensive for suborders of large discriminant), but we only need to compute their size.

Lemma 6.3 in [21] can be used to calculate the size of the isogeny class exactly. However it requires computing the lattice of suborders of the maximal order $\mathcal{O}_K$ which contain $\mathbb{Z}[\pi, \overline{\pi}]$ which is already too expensive. In practice, it is enough to estimate the size of the isogeny class using only the factorization of the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ and a calculation involving only the order $\mathbb{Z}[\pi, \overline{\pi}]$.

We compute using the proof of Lemma 6.3 in [21]

$$\#\mathfrak{C}(\mathbb{Z}[\pi, \overline{\pi}]) = \frac{c\#\operatorname{Cl}(\mathbb{Z}[\pi, \overline{\pi}])\operatorname{Reg}(\mathbb{Z}[\pi, \overline{\pi}])}{2\#\operatorname{Cl}(\mathbb{Z}[\pi + \overline{\pi}])\operatorname{Reg}(\mathbb{Z}[\pi + \overline{\pi}])}$$

where $c$ is the size of the co-kernel of the norm application from the class group of $\mathbb{Z}[\pi, \overline{\pi}]$ to the narrow class group of $\mathbb{Z}[\pi + \overline{\pi}]$. In the following we will use $c = 1$ in order to have a lower bound for $\#\mathfrak{C}(\mathbb{Z}[\pi, \overline{\pi}])$. Moreover Equation (6.1) of [21] give us

$$\#\operatorname{Cl}(\mathbb{Z}[\pi, \overline{\pi}])\operatorname{Reg}(\mathbb{Z}[\pi, \overline{\pi}]) = \#\operatorname{Cl}(O_K)\operatorname{Reg}(O_K)(\widehat{O}_K^* : \widehat{\mathbb{Z}}[\pi, \overline{\pi}]^*).$$

If $I$ is the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, we can compute $(\widehat{O}_K^* : \widehat{\mathbb{Z}}[\pi, \overline{\pi}]^*)$ as $((\mathcal{O}_K/I)^* : (\mathbb{Z}[\pi, \overline{\pi}]/I)^*)$. It is easy to compute $\#(\mathbb{Z}[\pi, \overline{\pi}]/I)^*$; since it is a torsion group and $I$ is prime to $p = (\mathbb{Z}[\pi, \overline{\pi}] : \mathbb{Z}[\pi])$ it is equal to $\prod_\ell \#(\mathbb{Z}[\pi]/I)_\ell^*$. Now if $\ell^e$ is a prime power of $I$, and $\chi_\pi = \prod P_i^{e_i}$ the factorization of $\chi_\pi \mod \ell$ then

$$\#(\mathbb{Z}[\pi]/I)_\ell^* = \prod_i (\ell^{\deg P_i e e_i} - \ell^{\deg P_i (e e_i - 1)}).$$

Likewise, we can compute $\#(\mathcal{O}_K/I)_\ell^*$ by looking at the decomposition of $\ell$ in $\mathcal{O}_K$.

Now we use the following estimate: for each divisor $d$ of the index $I$, the contributions of orders $O$ such that $[O : \mathbb{Z}[\pi, \overline{\pi}]] = d$ to curves in the isogeny class is $\#\mathfrak{C}(\mathbb{Z}[\pi, \overline{\pi}])/d$. So we estimate the number of curves as

$$\sum_{d | [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]} \#\mathfrak{C}(\mathbb{Z}[\pi, \overline{\pi}])/d$$

(for $d$ not divisible by a $\ell$ where we can't go up). We will see in Section 10 how this estimation compares to some real examples.

7.3. **Estimating the probability of going up.** In practice, we are not interested in the size of the isogeny class, but in the size of the curves in the isogeny class from where we can go up. From numerous experiment, we have observed that most of the cases where we can't go up for a particular $\ell$ is because there exist no rational $(\ell, \ell)$-isogenies at all. But we can easily detect this case by using Proposition 30.

We can thus estimate the number $\mathbf{C}$ of curves from where we can go up as

$$\sum_{d | [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]} \#\mathfrak{C}(\mathbb{Z}[\pi, \overline{\pi}])/d$$

but where we restrict the divisors $d$ to be such that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]/d$ is not divisible by a prime $\ell$ where we can't go up.

Now if $\mathbf{C}_0$ is the estimated number of curves in the isogeny class, we estimate that we need $\mathbf{C}_0/\mathbf{C}$ going up tries before finding a going up phase, and such of endomorphism ring computation. Now we keep the prime $p$ if $\mathbf{C}_0$ is not too small, and if the cost of doing all these endomorphism ring computation is at most the time needed to find a curve in the isogeny class where we can go up. (As we will see

in the complexity analysis, the endomorphism ring computation is not the dominant phase, so in practice this condition is almost always satisfied).

7.4. **A dynamic selection of primes.** When we select a prime $p$, we hope that the size $\mathbf{C}_0$ of the isogeny class (or more precisely the size $\mathbf{C}$ of curves where we can go up) is approximatively the average size $\Theta(p^{3/2})$ of an isogeny class. However, for small primes $p$, even if the isogeny class is small, it could be worth it to go through all $p^3$ curves corresponding to $p$ rather than through $q^{3/2}$ curves of a larger prime $q$.

So we use a dynamic approach: for each new CRT prime $p$, we compute $\mathbf{C}/p^3$, the expected probability of finding a curve where we can go up. We also look at the corresponding probabilities for the previous discarded CRT prime, and we use the prime giving the maximum probability provided it is more than $1/16p^{-3/2}$.

## 8. CRT lifting of the coefficients

8.1. **Denominators via Bruinier-Yang formula.** We note that contrary to the elliptic curve case, the coefficients of the class polynomials are rationals, not integers.

The CRT algorithm terminates when the lifted class polynomials (with denominators cleared) are constant from one CRT prime to the next. The probability that the class polynomials are correct when this happens was estimated in [14, Remark 7.2].

We estimate the denominators using the Bruinier-Yang conjectural formula [6] (proved only for special cases [31, 32]), with minor adjustments from [17], and adapting it to the fact that we use invariants from [27, Appendix 3], which alters the denominator formulas by small powers of 2. A formula for the factorization of the denominators which holds for general primitive quartic CM fields was recently given in [19] which produces a multiple of the denominators when allowing for cancellation with the numerators and the case where $K^+$ does not have class number 1. As in [12, Theorem 3], we can multiply by the denominators and then use the CRT to reconstruct the polynomials.

We note that since we are using Streng's invariant, the power of the $I_{10}$ in the denominators is $1, 2, 2$ rather than $6, 4, 4$ but we have to multiply them by the constant factors $2^7, 2^5, 2^{14}$.

8.2. **The CRT.** In the cyclic case, we compute the class polynomials modulo small integer primes, and we use the CRT to get the result modulo the product $P$ (the "precision") of these small primes. Once the precision is enough, we can recover the polynomials modulo $\mathbb{Z}$, by lifting each coefficient to $\mathbb{Z}$ in the interval $[-P/2, P/2]$.

In the dihedral case, the primes are in $\mathcal{O}_{K_\Phi^+}$, and so is the precision ideal $P$. Here we explain how to lift a coefficient $x$ mod $P$ to $\mathcal{O}_{K_\Phi^+}$. Take the Minkowski embedding of a lift of $x$, and find the closest vector $c_x$ in the lattice associated to $P$ in the Minkowski embedding. Then $c_x$ corresponds to an element of the ideal $P$, and our final lift is $x - c_x$. We note that the lattice is of rank 2, so we can directly compute the closest vector rather than doing an LLL approximation.

8.3. **Lifting without denominators.** We note that in the dihedral case, the denominator from the formulas in [6, 17, 19] is too large, as it takes into account both CM types. This increases the size of the coefficients we compute, so that using those denominator formulas does not actually give better results than doing a rational reconstruction directly.

With the notation from above, from $x$ mod $P$ we want to do a rational lift of $x$. This time we embed the lattice associated to $P$ into the lattice of rank 3 obtained by adjoining the vector $[Cx_1, Cx_2, C]$ where $x_1$ and $x_2$ are the two real embeddings of (a lift of) $x$ and $C$ is a constant accounting for how skewed we expect the size of the denominator to be compared to the numerators. A minimal vector in this lattice will correspond to an element $N = c + Dx$ where $c \in p$ and $D$ is an integer. We then take $N/D$ as our lift for $x$.

This solution requires the precision to be the sum of the bit sizes of the numerators and denominator, so it can be even better than using the denominator formulas for small denominators where there may be cancellation with the numerators.

8.4. **Finding irreducible factors.** Computing irreducible factors of the class polynomials directly allows to recover them faster since they have smaller coefficients, so that we need less precision.

We know that the orbits under the action of the type norm give the irreducible factors of the class polynomials (or more precisely the irreducible components of the CM locus) [27, Chapter 3]. It is easy to compute these orbits modulo each CRT prime $p$ using the tools from Section 6. However we need to be able to glue the correct orbits together when doing the CRT. For this we can use the "trace trick" from [13].

## 9. Complexity

In this section, we analyze the effect of the going up algorithm of Section 5 (which we will also call the vertical step since it deals with abelian varieties which have endomorphism rings which are different orders in $K$) and the effect of finding all other maximal curves from one maximal curve from Section 6 (the horizontal step) to the asymptotic complexity. Most of the discussion is heuristic.

We begin with a quick reminder of the rough complexity analysis of the CRT method in the elliptic curve case, where $K$ is a quadratic imaginary field. There is only one class polynomial $H$, whose degree is the class number of $\mathcal{O}_K$, and classical bounds give that $\deg H = O(\sqrt{\Delta})$ where $\Delta$ is the discriminant of $\mathcal{O}_K$. Likewise, the coefficients of $H$ have size $\tilde{O}(\sqrt{\Delta})$. So the whole class polynomial is of size $\tilde{O}(\Delta)$.

Each CRT prime $p$ gives $\log(p)$ bits of information, so neglecting logarithmic factors, we need $\sqrt{\Delta}$ primes. CRT primes split completely in the Hilbert class field of $K$, whose Galois group is $\mathrm{Cl}(\mathcal{O}_K)$, so by the Cebotarev theorem the density of CRT primes is roughly $1/\#\,\mathrm{Cl}(\mathcal{O}_K) \simeq 1/\Delta$. Neglecting logarithmic factors again, the biggest prime $p$ is of size $\tilde{O}(\Delta)$.

Now there are $O(p)$ isomorphism classes of elliptic curves, and $\tilde{O}(\sqrt{\Delta})$ maximal curves, so one is found in time $\tilde{O}(p/\sqrt{\Delta}) = \tilde{O}(\sqrt{p})$. Once one maximal curve is found, all others can be obtained using isogenies of degree logarithmic in $\Delta$, so one can recover all maximal elliptic curves over $\mathbb{F}_p$ in time $\tilde{O}(\sqrt{p}) = \tilde{O}(\sqrt{\Delta})$.

We need $\sqrt{\Delta}$ CRT primes, so the total cost is $\tilde{O}(\Delta)$. The CRT reconstruction can be done in quasi-linear time too, so in the end the algorithm is quasi-linear, even without using a vertical step. Not using the horizontal step gives a complexity of $\tilde{O}(\Delta^{3/2})$.

In genus 2, let $\Delta_0 = \Delta_{K^+/\mathbb{Q}}$ and $\Delta_1 = N_{K^+/\mathbb{Q}}(\Delta_{K/K^+})$, so that $\Delta = \Delta_{K/\mathbb{Q}} = \Delta_1\Delta_0^2$. Then the degree of the class polynomials is $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ while the height of their coefficients is bounded by $\tilde{O}(\Delta_0^{5/2}\Delta_1^{3/2})$ ([27, Section II.9], [16]). In practice,

we observe [27, Appendix 3] that they are bounded by $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ and we will use this bound in the following. According to [5, Section 6.4], the smallest prime is of size $\tilde{O}(\Delta_0\Delta_1)$. We need $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ CRT primes, and an analysis using [18] as in [2, Lemma 5.3] shows that the largest prime is also $\tilde{O}(\Delta_0\Delta_1)$. We remark that the sieving phase does not affect the size of the largest prime (apart from the constant in the big $O$) as long as we sieve a positive density of CRT primes.

For the horizontal step, the isogeny computation involves primes of size logarithmic in $\Delta$, so the cost of this step is quasi-linear in the number $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ of maximal curves. This is under the assumption from Section 6. Without this assumption, what we know is that for each ideal $I$ in $\mathcal{O}_{K_\Phi}$ of norm prime to $p$, the element $(TN(I), N(I))$ is an element of the Shimura class group whose action is given by a maximally isotropic kernel. In the horizontal step, we can then compute the action of $TN(\mathrm{Cl}(\mathcal{O}_{K_\Phi}))$ by isogenies of size logarithmic in $\Delta$. By Lemma 6.5 of [5], the cofactor is bounded by $2^{6w(D)+1}$, where $w(D)$ is the number of prime divisors of $D$. This gives a bound on the number of horizontal isogeny steps we need to take. As remarked in [5], outside a zero-density subset of very smooth integers, $w(n) < 2\log\log n$ so the corresponding factor can be absorbed into the $\tilde{O}$-notation.

On the contrary, the complexity of the endomorphism ring computation and the going up phase involves the largest prime power dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. According to Proposition 6.1 of [14] we have that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] \leq \frac{16p^2}{\sqrt{\Delta}}$. For the size of the CRT prime we are considering, we see that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \tilde{O}(\Delta_0\Delta_1^{3/2})$. We fix $\epsilon = 1/2$. Assuming that the index is uniformly distributed, [11] showed that there is a positive density of CRT primes where the largest prime power dividing the index is $O(\Delta_0^{\epsilon/100}\Delta_1^{\epsilon/100})$. By the complexity analysis of Sections 4.6 and 5.3, we see then that there is a positive density of primes where these algorithms take time at most $O(\Delta_0^{\epsilon}\Delta_1^{\epsilon})$.

We then let $p = \tilde{O}(\Delta_0\Delta_1)$ be a CRT prime. There are $O(\sqrt{p})$ maximal curves, so we expect the isogeny class to be of size $\Theta(p^{3/2})$ (see Heuristic 6.6 in [5]), and there are $p^3$ isomorphism classes of curves. The original CRT algorithm of [12, 14] looped through all $p^3$ curves and tested if the endomorphism ring is maximal. This takes $\tilde{O}(\Delta_0^3\Delta_1^3 + O(\Delta_0^{3/2+\epsilon}\Delta_1^{3/2+\epsilon})$ per CRT prime, for a total cost (since $\tilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ CRT primes are needed) of $\tilde{O}(\Delta_0^{7/2}\Delta_1^{7/2})$ with our choice of $\epsilon$.

The approach of [5] is to find only one of them and use horizontal isogenies to find the others. With the improvements proposed in this paper (using all horizontal isogenies and not only those coming from the type norm, and the improved endomorphism ring computation): we find a cost of $\tilde{O}(\Delta_0^{5/2}\Delta_1^{5/2}) + O(\Delta_0^{3/2+\epsilon}\Delta_1^{3/2+\epsilon})$ per CRT prime. The total cost is then $\tilde{O}(\Delta_0^3\Delta_1^3)$.

With our method, we need to find a curve in the isogeny class where the going up step yields a maximal curve. Finding a curve in the isogeny class takes time $O(p^{3/2})$. If $X$ is the number of going up steps we need to try on average, the cost per CRT prime is then $\tilde{O}(X(\Delta_0^{3/2}\Delta_1^{3/2} + \Delta_0^{\epsilon}\Delta_1^{\epsilon}))$. At best, $X = O(1)$, and we have a total cost of $\tilde{O}(\Delta_0^2\Delta_1^2)$ from CRT primes. So at best we have a quasi-quadratic complexity, while the CRT itself is quasi-linear, so is negligible. We see that we are still far from quasi-linearity achieved by the analytic method. At worst, $X = O(p)$ (number of random tries in the isogeny class until we find a maximal one directly), and we recover the quasi-cubic complexity of the previous method.

To improve the complexity, there are two possibilities: the first is to increase the probability of success of the going up method. This requires an algorithm to compute isogenies with cyclic kernels. But even with that, we achieve at most quasi-quadratic complexity because the size of the isogeny class is too small compared to the size of the search space. This is the case because the algorithm computes the class polynomials (a scheme of dimension 0) directly from the moduli space of dimension 3 of all abelian surfaces. In contrast, in the elliptic curve case, the algorithm searches a space of dimension 1 for elements of a space of dimension 0. It would be interesting to find convenient subspaces of the moduli space of smaller dimension, and to work over them. One example would be to use Humbert surfaces, which are of dimension 2, and Gundlach invariants, as proposed in [20].

## 10. Examples

10.1. **Improvements from the going up phase.** We first look at improvements due solely to the going up phase. For that we look at Galois examples with class number one; so there is only one maximal curve and the algorithm from Section 6 is not used. We compare the timings to the three examples in [14, Section 9], so to use (almost) the same CRT primes we deactivate the dynamic approach in the prime selection.

The timings are given in Tables 1, 2 and 3. The first column indicates the CRT prime used. The second one the $\ell^d$-torsion subgroups required to compute if a curve is maximal, the $\ell$ is in bold if there is no $(\ell,\ell)$-isogeny, so we can't go up for this $\ell$ (and it is in italic if there exist $(\ell,\ell)$-isogenies, but they are too expensive to compute). The third column give the corresponding degree where the points of these subgroups live. The fourth column indicates the total number of curves in the isogeny class (this is found by the algorithm from [14] since they need to go through all the curves), while the fifth give the estimate we compute of the number of curves from where we can go up (and the number in parenthesis give our estimate of the total number of curves in the isogeny class). The last two column give the timings of the old and new algorithm, split in "Time exploring curves" + "Time spent computing endomorphism rings/Time spent going up".

We note that we spend much less time exploring curves with the new algorithm which is the whole purpose of the going up part. But we also note that even through our going up phase is more complicated, it is still less costly than the computation of the endomorphism rings of the old algorithm. This is due to the improvements described in Section 4.

Note that much less time is spent exploring curves with the new algorithm due to the going up algorithm. Also note that, even though the going up phase is more complicated, it is still less costly than the computation of the endomorphism rings in the old algorithm, due to the improvements described in Section 4 and the fact that the new version calls it less often.

The trade-offs in the going up step depend on the discriminant of the CM field $K$. The more CRT primes we need, the bigger the isogenies and the bigger the degrees in the endomorphism ring computations we allow. Note that computing $(\ell,\ell)$-isogenies requires $\tilde{O}(\ell^2)$ operations in the field where the points of the kernel are defined when $\ell$ is congruent to 1 (mod 4), but $\tilde{O}(\ell^4)$ when $\ell$ is congruent to 3 (mod 4). So in the above example, we computed the $(109, 109)$-isogenies faster than the $(23, 23)$-isogenies.

| $p$ | $l^d$ | $\alpha_d$ | # 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 |
| | | | | | $3162s$ | $13s$ |

TABLE 1. Computing the class polynomial for $K = \mathbb{Q}(i\sqrt{2+\sqrt{2}})$, $\mathfrak{C}(O_K) = \{0\}$.

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

| $p$ | $l^d$ | $\alpha_d$ | # 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 | $3147+16.5$ | - |
| | | | | | $6969s$ | $114s$ |

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

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

10.2. **Dihedral examples.** Here we illustrate our new CRT algorithm for dihedral fields, for $K = \mathbb{Q}(X)/(X^4 + 13X^2 + 41)$ with $\mathfrak{C}(K) \simeq \{0\}$.

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 use the invariants from [27, Appendix 3]. We get $H_1 = 256X - 2030994 + 56133\alpha$, $H_2 = 128X + 12637944 - 2224908\alpha$, $H_3 = 65536X - 11920680322632 + 1305660546324\alpha$ where $\alpha$ is a root of $X^2 - 3534X + 177505$, so that $O_{K_0^+} = \mathbb{Z}[\alpha]$. This computation took 1401 seconds,

| $p$ | $l^d$ | $\alpha_d$ | # 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 | $2, \mathbf{5}$ | $1, 12$ | 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 | $7, \mathbf{13}$ | $3, 21$ | 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 | $5, 109$ | $24, \mathit{5940}$ | - | - | - | - |
| 199 | $\mathbf{5}^2$ | 60 | 37 | 2 (39) | $10440 + 35.1$ | - |
| 223 | $2, \mathit{23}$ | $1, 11$ | 1058 | 39 (914) | $10440 + 35.1$ | - |
| 227 | 109 | *1485* | - | - | - | - |
| 233 | $5, 7, \mathbf{13}$ | $8, 3, 28$ | 735 | 55 (770) | $11580 + 141.6$ | $88.3 + 29.4$ |
| 239 | $7, 109$ | $6, \mathit{297}$ | - | - | - | - |
| 257 | $3, 7, \mathbf{13}$ | $4, 6, 84$ | 1155 | 109 (1521) | $17160 + 382.8$ | - |
| 313 | $3, \mathbf{13}$ | $1, 14$ | ? | 146 (2035) | - | $165 + 14.7$ |
| 373 | $5, 7$ | $6, 24$ | ? | 312 | - | $183.4 + 3.8$ |
| 541 | $2, 7, \mathbf{13}$ | $1, 3, 14$ | ? | 294 (4106) | - | $91 + 5.5$ |
| 571 | $3, \mathbf{5}, 7$ | $2, 6, 6$ | ? | 1111 (6663) | - | $96.6 + 3.1$ |
| | | | | | 56585s | 776s |

$$H_1 = 244140625X - 2614061544410821165056,$$
$$H_2 = 390625X + 14651024316825600,$$
$$H_3 = 390625X + 5076177577574400.$$

TABLE 3. The class polynomial for $K = \mathbb{Q}(i\sqrt{29 + 2\sqrt{29}})$, $\mathfrak{C}(O_K) = \{0\}$. (The new algorithm also skipped the primes 277, 281, 349, 397, 401, 431, 487, 509, 523.)

so in this case, the speedup due to using better invariants and computing over the real subfield is more than 4-fold.

10.3. **CRT for non principal fields.** Here is a bigger example. To our knowledge, this is the first example of class number greater than one computed by the CRT method (we see that we are still far away from the analytic method where the latest version can compute class polynomials with degrees in the thousands):

- $K = \mathbb{Q}(X)/(X^4 + 238X^2 + 833)$ cyclic. $\mathfrak{C}(K) \simeq \mathbb{Z}/2\mathbb{Z}$ is generated by $(7, 7)$-isogenies.
- Primes used: 19 , 59 , 67, 83, 149, 191, 223, 229, 239, 257, 349, 463, 557, 613, 661, 733, 859, 1039, 1373, 1613, 1657, 1667, 1733, 1753, 1801, 1871, 1879, 2399, 3449, 3469, 3761, 3931, 4259, 4691, 5347, 5381, 6427, 6571, 6781.
- For $p \approx 6000$, we keep $p$ if we expect more than $\frac{p^{3/2}}{32} \approx 15 \times 10^6$ curves. At this size, it takes around 6 seconds to test 10000 curves, so around 2.5 hours are needed for $p$.
- Total time: 44062 second (not using the dynamic approach).
- We only give the first class polynomial:

$$H_1(X) = 16845120063354536424359491014628690731657228186228087100579542361282969 6X^2$$
$$+15858252869551393497069303119852348926972411909463014567206273563251802650749789064396 8X$$
$$-20148439779616498933576752193721158991703786695904651875585742599422503529550925413744 64.$$

## 11. Appendix: complexity of computing a $(\ell, \ell)$-isogeny

We recall the following proposition from [9]

**Proposition 30.** *The complexity of computing a $(\ell, \ell)$-isogeny between two abelian surfaces is $O(\ell^r)$ operations in the field $k$ where the points of the kernel of the isogeny live. We have $r = 2$ when $\ell \equiv 1 \mod 4$ and $r = 4$ when $\ell \equiv 3 \mod 4$.*

If $A$ is an abelian variety over a finite field $\mathbb{F}_q$, the following proposition we give a way to compute the maximum extension over which the points of a maximal isotropic kernel live.

**Proposition 31.** *Let $\chi_\pi$ be the quartic polynomial satisfied by the Frobenius element for a smooth irreducible genus 2 curve $C$ over $\mathbb{F}_p$ with simple, ordinary Jacobian $J(C)$. For a prime $\ell$ with $\ell \neq p$, if there exists an $\mathbb{F}_p$-rational $(\ell, \ell)$-isogeny, then $\chi_\pi$ factors as $\chi_\pi = P\overline{P} \pmod{\ell}$ (where $\overline{P}$ is the conjugate of $P$ under the action $\pi \to p/\pi$).*

*The order of $X$ in $\mathbb{Z}[X]/(\ell, P)$ give the degree in which the points of the corresponding Kernel live. In particular, if no such decomposition exist, then there is no $(\ell, \ell)$-isogeny.*

*Proof.* Let $K \subset A[\ell]$ be a maximally isotropic rational kernel. Then since $\pi$ stabilize $K$, if $P$ is the characteristic polynomial of $\pi$ restricted to $K$, then $P$ divides $\chi_\pi$. The cofactor is given by the characteristic polynomial of the action of the Verschiebung $p/\pi$ on $K$, so the corresponding factor is $\overline{P}$. □

## References


[1] E. Bach *Explicit Bounds for Primality Testing and Related Problems*, Mathematics of Computations, **55**, No. 191, (1990) 335–380.

[2] J. Belding, R. Bröker, A. Enge, K. Lauter, *Computing Hilbert class polynomials*, ANTS VIII, Springer LNCS **5011** (2008), 282–295.

[3] G. Bisson, R. Cosset, D. Robert, *AVIsogenies (Abelian Varieties and Isogenies)*, Magma package for explicit isogenies between abelian varieties, `http://avisogenies.gforge.inria.fr`.

[4] G. Bisson, *Endomorphism Rings in Cryptography*, PhD Thesis (2011).

[5] R. Bröker, D. Gruenewald, K. Lauter, *Explicit CM-theory for level 2-structures on abelian surfaces*, Algebra and Number Theory, Vol. 5 (2011), No. 4, 495–528. DOI: 10.2140/ant.2011.5.495.

[6] J. Bruinier, T. Yang, CM-values of Hilbert modular functions, Invent. Math., **163**, no. 2, (2006) 229–288.

[7] Robert Carls, David Kohel, David Lubicz, *Higher dimensional 3-adic CM construction*, J. of Algebra, 319-4 (2008), 971–2006.

[8] Robert Carls, David Lubicz, *A p-adic quasi-quadratic point counting algorithm*, *Int. Math. Res. Not.*, 2008.

[9] R. Cosset, D. Robert, *An algorithm for computing $(\ell, \ell)$-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus* 2, Preprint 2011, `http://eprint.iacr.org/2011/143`.

[10] J.M. Couveignes, *Linearizing torsion classes in the Picard group of algebraic curves over finite fields*, Journal of Algebra, **321**, no. 8, (2009) 2085–2118.

[11] K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude*, Ark. Mat. Astr. Fys. 22A (1930), 1–14.

[12] K. Eisenträger, K. Lauter, *A CRT algorithm for constructing genus 2 curves over finite fields*, In: Arithmetic, Geometry and Coding Theory AGCT-10 (2005), Séminaires et Congrès **21**, Société Mathématique de France (2009) 161–176. `http://arxiv.org/abs/math.NT/0405305`

[13] A. Enge, A.V. Sutherland, *Class invariants by the CRT method*, ANTS IX: Proceedings of the Algorithmic Number Theory 9th International Symposium, Nancy (2010), 142–156.

[14] D. Freeman, K. Lauter, *Computing endomorphism rings of Jacobians of genus 2 curves over finite fields*, in Algebraic Geometry and its Applications, World Scientific (2008), 29–66.

[15] P. Gaudry, T. Houtmann, D. Kohel, C. Ritzenthaler, and A. Weng, *The 2-adic CM method for genus 2 curves with application to cryptography*, Asiacrypt 2006 (Shanghai) Lect. Notes in Comp. Sci., 4284, 114–129, Springer-Verlag, 2006.
[16] E. Goren, K. Lauter, *Genus 2 curves with complex multiplication*, International Mathematics Research Notices (2011), 75 pp. doi: 10.1093/imrn/rnr052.
[17] H. Grundman, J. Johnson-Leung, K. Lauter, A. Salerno, B. Viray, E. Wittenborn, *Igusa class polynomials, embeddings of quartic CM fields, and arithmetic intersection theory*, WIN-Women in Numbers: Research Directions in Number Theory, Fields Institute Communications Series, Volume 60 (2011) 35–61.
[18] J. C. Lagarias, A. M. Odlyzko, *Effective versions of the Chebotarev density theorem*, Algebraic number fields: L-functions and Galois properties (Proc. Sympos., Univ. Durham, Duram, 1975), Academic Press, 1977, pp. 409–464.
[19] K. Lauter, B. Viray, *An arithmetic intersection formula for denominators of Igusa class polynomials*, Preprint, 2011.
[20] K. Lauter, T. Yang, *Computing genus 2 curves from invariants on the Hilbert moduli space*, Journal of Number Theory, Elliptic Curve Cryptography Volume **131**, Issue 5 (2011), 936–958.
[21] H. W. Lenstra, Jr., J. Pila, C. Pomerance, *A hyperelliptic smoothness test II*, Proc. London Math. Soc., (3) **84** (2002), 105–146.
[22] D. Lubicz, D. Robert, *Computing isogenies between abelian varieties*, to appear in Compositio Mathematica, `http://arxiv.org/pdf/1001.2016`.
[23] J.-F. Mestre, *Construction de courbes de genre 2 à partir de leurs modules*, in Effective methods in algebraic geometry, Birkhäuser Progr. Math. **94** (1991), 313–334.
[24] D. Robert, *Theta functions and applications in cryptography*, PhD Thesis, Université Henri-Poincarré, (2010).
[25] G. Shimura, *Abelian varieties with complex multiplication and modular functions*, Princeton Mathematical Series **46**, Princeton University Press New Jersey, 1998.
[26] A.-M. Spallek, *Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen*, PhD thesis, Universität Gesamthochschule Essen, 1994.
[27] M. Streng, *Complex multiplication of abelian surfaces*, PhD-thesis, Universiteit Leiden, 2010.
[28] A. Sutherland, *Computing Hilbert class polynomials with the Chinese Remainder Theorem*, Mathematics of Computation **80** (2011), pp. 501-538.
[29] P. van Wamelen. *Examples of genus two CM curves defined over the rationals*, Math. Comp., 68 (1999), 307–320.
[30] A. Weng, *Constructing hyperelliptic curves of genus 2 suitable for cryptography*, Math. Comp. 72 (2003), 435–458.
[31] T. Yang, *An arithmetic intersection formula on Hilbert modular surfaces*, *Amer. J. Math.*, 132 (2010), 1275–1309.
[32] T. Yang, *Arithmetic intersection on a Hilbert modular surface and the Faltings height*, Preprint, 2007.

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence cedex, France
*E-mail address*: `damien.robert@inria.fr`

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA.
*E-mail address*: `klauter@microsoft.com`