Contrat de recherches numéro 2010 42 0349
Rapport numéro 8

Computing class polynomials in genus 2

Rapport DGA

Andreas Enge and Damien Robert

26 April 2013

Inria Bordeaux Sud-Ouest,
200 avenue de la Vieille Tour
33405 Talence Cedex

andreas.enge@inria.fr
damien.robert@inria.fr

Part of this report is taken from [ET13] and [LR12]. We thank our respective co-authors Emmanuel Thomé and Kristin Lauter for allowing us to reuse the material.

Contents

1 Complex multiplication in genus 1 and 2

1.1 Public key cryptography

The discrete logarithm problem is at the heart of modern public key cryptography. It requires finding mathematical groups in which the discrete logarithm problem is hard, whereas exponentiation is efficient to compute.

Currently, the best such groups come from algebraic geometry, namely from elliptic curves (abelian varieties of dimension 1) and Jacobians of hyperelliptic curves of genus 2 (abelian varieties of dimension 2) over finite fields. Abelian varieties of greater dimension than 2 have been proposed, but due to faster attacks than the generic ones [GTT+07], they are essentially ruled out from practical use.

To instantiate the discrete logarithm problem, one needs to find an elliptic curve or an abelian variety of dimension 2 over a finite field $\mathbb{F}_q$ whose cardinality is divisible by a large prime factor, the generic algorithm for discrete logarithms taking time proportional to the square root of the largest prime factor.

There are two approaches for finding such suitable varieties. The first one takes a random variety and uses a point counting algorithm until a suitable instance is found. Note that both in dimension 1 and 2 polynomial time algorithms are available [Sch85; Pil90]. However, while point counting in genus 1 is nowadays very fast because of improvements described in [Atk88; Elk97], this is not quite the case yet in genus 2 for a random curve (compare [Sut] to [GS08]), although random curves have definitely becomes an option in genus 2 as well. We note that it has been recently proposed not to take random genus 2 curves, but to draw them in a family with real multiplication, in which case there are faster point counting algorithms available [GKS11].

The second approach, which is the main subject of this report, is to use the theory of complex multiplication for constructing a variety with a prescribed number of points. The idea is to construct the moduli space of all abelian varieties (of dimension 1, resp. 2) having for endomorphism ring a fixed order of a CM field K (of degree 2, resp. 4). This moduli space is of dimension 0, so can be represented as the zero locus of polynomials in the moduli coordinates. These are the so called class polynomials. The moduli field of these abelian varieties is a number field H (which is related to K), so it make sense to consider reduction modulo a prime p. The theory of complex multiplication (from Deuring [Deu58] for elliptic curves, and from Taniyama-Shimura-Weil for abelian varieties [Shi98]) then describes the action of the Frobenius endomorphism on the reduction. In particular, one can recover the number of points.

This yields the following method: Fix an order $\mathcal{O}$ in a CM field K. In this report, we will assume that $\mathcal{O}$ is the maximal order $\mathcal{O}_K$ for simplicity, but this is not a strict requirement. Compute the class polynomials for $\mathcal{O}$. Find a prime p such that the associated Frobenius corresponds to a Weil polynomial with certain properties (typically we want the varieties to be defined over $\mathbb{F}_p$ and have a large prime dividing their order, but we may also require a small embedding degree for use in pairing-based cryptography). Reduce the class polynomial modulo p and find a root; this root corresponds to the moduli invariant of a variety with CM by $\mathcal{O}$ reduced modulo p. From these invariants (the j-invariant in dimension 1 or the Igusa invariants in dimension 2) it is then easy to recover the corresponding elliptic curve or hyperelliptic curve of genus 2 (in the latter case using the Mestre-

Cardona-Quer [Mes91; CQ05] algorithm).

We note that the complexity of the method based on taking random curves depends on the size of the finite field we want to work with. By contrast, the main parameter for the complexity of the CM method is in the size of the class polynomials, which is related to the discriminant of the field K.

1.2 Computing class polynomials in genus 1

1.2.1 CM theory for elliptic curves

Let $E/\mathbb{C}$ be an elliptic curve with CM by an order $\mathcal{O}$ of an imaginary-quadratic field K. The isomorphism class of E is represented by its moduli parameter j, the j-invariant.

Theorem 1.2.1 (Kronecker, Deuring) : *The j-invariant $j(E)$ of E is an algebraic number, and $K(j(E))$ is the ring class field $\mathfrak{H}_{\mathcal{O}}$ of $\mathcal{O}$. (In particular, if $\mathcal{O}$ is maximal, $\mathfrak{H}_{\mathcal{O}}$ is the Hilbert class field).*

Moreover, the dimension 0 moduli space of elliptic curves with CM by $\mathcal{O}$ is a torsor under the Galois group $\operatorname{Gal}(\mathfrak{H}_{\mathcal{O}}/K) \simeq \operatorname{Cl}(\mathcal{O})$, where $\mathfrak{a} \in \operatorname{Cl}(\mathcal{O})$ acts by an isogeny of degree $N(\mathfrak{a})$. In particular, the minimal polynomial of $j(E)$ over K is given by the class polynomial

$$H = \prod_{C} (X - j(C))$$

where the product is taken over all elliptic curves C with CM by $\mathcal{O}$.

The class polynomial H is monic and lies in $\mathbb{Z}[X]$, so the field $\mathfrak{H}_{\mathcal{O}}^0 = \mathbb{Q}(j(E))$ is of dimension $\deg H$ over $\mathbb{Q}$, linearly independent of K, and $\mathfrak{H}_{\mathcal{O}} = \mathfrak{H}_{\mathcal{O}}^0 K$.

Deuring also described the reduction of CM elliptic curves modulo a prime p.

Theorem 1.2.2 : *If $p \nmid \Delta_{\mathcal{O}}$ is a non-ramified prime, then p is a prime of (potentially) good reduction. Let $\mathfrak{P}$ be a prime above p in $\mathfrak{H}_{\mathcal{O}}$. If p is inert or ramified in K, $E_{\mathfrak{P}}$ is supersingular. Otherwise p splits in K, $E_{\mathfrak{P}}$ is ordinary, and its endomorphism ring is the minimal order containing $\mathcal{O}$ of index prime to p.*

Moreover, let $\mathfrak{p} = \mathfrak{P} \cap K$; since the Artin symbol $\left(\frac{\mathfrak{P}}{\mathfrak{p}}\right)$ is given by the class of $\mathfrak{p} \in \operatorname{Cl}(\mathcal{O})$, we see that the elliptic curve E reduces to $\mathbb{F}_{\mathfrak{p}^n}$ where n is the order of the ideal $\mathfrak{p}$. We then have that $\mathfrak{p}^n$ is principal, and we can write $\mathfrak{p}^n = (\pi)$ where $\pi\overline{\pi} = N(\mathfrak{p})^n$ (this condition determines π up to roots of unity). Then $\pi \in \mathcal{O}$ corresponds to the action of the Frobenius of $\mathbb{F}_{\mathfrak{p}^n}$ on $E_{\mathfrak{P}}$.

Remark 1.2.3 : If p is ramified in $\mathcal{O}$, then E has supersingular reduction modulo p. ◇

Reciprocally, if $E/\mathbb{F}_q$ is an ordinary elliptic curve, the couple $(E, \operatorname{End}(E))$ can be lifted over $\mathbb{Q}_q$, so we have the following corollary:

Corollary 1.2.4 : *If $E/\mathbb{F}_q$ is an ordinary elliptic curve, then $\operatorname{End}(E)$ is an order in $K = \mathbb{Q}(\pi)$ of conductor prime to p. For every order $\mathcal{O}$ of K such that $\mathbb{Z}[\pi] \subset \mathcal{O}$, there exists a curve whose endomorphism ring is $\mathcal{O}$.*

Reciprocally, for every order $\mathcal{O}$ of discriminant a non-zero square modulo p, let n be the order of one of the primes above p in the class group of $\mathcal{O}$. Then there exists an (ordinary) elliptic curve E' over $\mathbb{F}_{p^n}$ with $\operatorname{End}(E') = \mathcal{O}$.

For computing the class polynomials in practice, there are three families of algorithm. In the following, we will assume that $\mathcal{O} = \mathcal{O}_K$ is the maximal order for simplicity. If Δ is the discriminant of $\mathcal{O}_K$, then classical bounds give that $\deg H = \widetilde{O}(\sqrt{\Delta})$ and that the coefficients of H have size $\widetilde{O}(\sqrt{\Delta})$. So the whole class polynomial is of size $\widetilde{O}(\Delta)$.

1.2.2 The complex analytic method

We fix an embedding φ of K into $\mathbb{C}$. If $\mathfrak{a}$ is an ideal of $\mathcal{O}_K$, then $\varphi(\mathfrak{a})$ is a lattice in $\mathbb{C}$, and the torus $\mathbb{C}/\varphi(\mathfrak{a})$ is an elliptic curve with CM by $\mathcal{O}_K$. The analytic method then computes a representative set for the class group $\mathrm{Cl}(\mathcal{O}_K)$, and for each ideal $\mathfrak{a}$ in this set compute the j-invariant (at a certain precision) of the lattice $\varphi(\mathfrak{a})$. These j-invariants are the roots of H. By using asymptotically fast algorithms [Eng09], one can then reconstruct H in quasi-linear time.

1.2.3 The p-adic lifting method

For the p-adic lift, one starts from an ordinary curve E defined over $\mathbb{F}_q = \mathbb{F}_{p^n}$ whose endomorphism ring is $\mathcal{O}_K$. One can use Theorem 1.2.2 to determine over which p and n one can search for such a curve. (Typically n is very small, so p will be of size $\widetilde{O}(\Delta)$.)

Then one can compute the canonical lift of E on $\mathbb{Q}_q$. The idea of using canonical lifts originates from Satoh for point counting over small characteristic [Sat00]. By using the modular polynomial of level p and a good basis for the field $\mathbb{F}_q$, such a lift can be computed in quasi-linear time in the precision [LL03; Gau04].

Once such a canonical lift is computed, one can use isogenies (via modular polynomials or Vélu's formulæ [Vél71]) to recover all other canonical lifts of elliptic curves with CM by $\mathcal{O}_K$ (since the Galois-group $\mathrm{Cl}(\mathcal{O}_K)$ acts by isogenies).

Under GRH, $\mathrm{Cl}(\mathcal{O}_K)$ is generated by ideals of small norm ($O(\log^2 \Delta)$), so recovering all the conjugate roots from the first lift can be done efficiently. Altogether this also gives a quasi-linear algorithm to compute H [Brö08].

The advantage of the p-adic method is that there is a better control of the precision loss than for the analytic method. In particular, it is easier to give guaranteed results.

1.2.4 The CRT method

The CRT method works by computing the class polynomial modulo many small primes, and then reconstructing the polynomials with rational coefficients (or modulo a much larger prime number) via the Chinese remainder theorem (respectively, the explicit CRT).

One way to compute the class polynomial H modulo p is to choose a prime p such that all elliptic curves with CM by $\mathcal{O}_K$ have good ordinary reduction to $\mathbb{F}_p$. Such a prime p will be called a CRT prime. By Theorem 1.2.2, p is a CRT prime if and only if p splits in K into principal ideals. Then it is only a matter of finding all elliptic curves over $\mathbb{F}_p$ with CM by $\mathcal{O}_K$, and reconstruct the class polynomial modulo p via its roots as for the analytic and p-adic case.

The idea of using the CRT to compute class polynomials was first presented in [BBE+08], where they used the following algorithm:

Algorithm 1.2.5 :

```
  INPUT: An imaginary quadratic field $K$, and a collection of CRT primes $P_K$ for $K$.
OUTPUT: The class polynomial $H(x)$ either in $\mathbb{Z}[X]$ or reduced modulo a prime $q$.
        1) Loop through CRT primes $p \in P_K$:
           (a) Enumerate elliptic curves $E$ over $\mathbb{F}_p$ until a curve with maximal endomorphism ring
               is found;
```

```
    (b) From a maximal curve $E$, use isogenies to compute all other maximal curves.
    (c) Reconstruct the class polynomial $H(X)$ modulo $p$ from the $j$-invariants of the set of
        maximal curves.
2) Recover $H(X)$ in $\mathbb{Z}[X]$ or modulo $q$ using the (explicit) CRT method once we have
   computed $H(X)$ modulo $p$ for enough primes $p$.                                        ♢
```

By [LO77], the smallest CRT prime is of size $\widetilde{O}(\Delta)$. Each CRT prime p gives $\log(p)$ bits of information, so neglecting logarithmic factors, we need about $\widetilde{O}(\sqrt{\Delta})$ 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 $1/\#\mathrm{Cl}(\mathcal{O}_K) \simeq 1/\sqrt{\Delta}$. Neglecting logarithmic factors again, we therefore expect the biggest prime p to be of size $\widetilde{O}(\Delta)$, and this is indeed what [LO77] gives.

Now there are $O(p)$ isomorphism classes of elliptic curves, and (under GRH) $\widetilde{\Omega}(\sqrt{\Delta})$ curves with $\mathcal{O}_K$ as endomorphism ring, so one of them is found in time $\widetilde{O}(p/\sqrt{\Delta}) = \widetilde{O}(\sqrt{p})$. Once one maximal curve is found, all others can be obtained using isogenies of degree logarithmic in Δ, so one can recover all maximal elliptic curves over $\mathbb{F}_p$ in time $\widetilde{O}(\sqrt{p}) = \widetilde{O}(\sqrt{\Delta})$.

We need $\sqrt{\Delta}$ CRT primes, so the total cost is $\widetilde{O}(\Delta)$. The CRT reconstruction can be done in quasi-linear time too, so in the end the algorithm is quasi-linear. We note the importance of using isogenies once we have a curve with CM by $\mathcal{O}_K$, otherwise the complexity would have been $\widetilde{O}(\Delta^{3/2})$.

To find an elliptic curve over $\mathbb{F}_p$ we can go through all j-invariants or take random elliptic curves E and test if E is in the isogeny class corresponding to the Weil polynomial of $\pi \in \mathcal{O}_K$ (where π is given by Theorem 1.2.2). If this is the case, then we know that $\mathrm{End}(E)$ is an order between $\mathcal{O}_K$ and $\mathbb{Z}[\pi]$. We can then test if the endomorphism ring of E is maximal, that is if E lives in the crater of the isogeny volcano [Koh96; FM02].

We note that there is some latitude in the choice of the CRT primes. In [BBE+08] the authors suggest looking for CRT primes such that the index $[\mathcal{O}_K : \mathbb{Z}[\pi]]$ is small, so that a good proportion of the curves in the isogeny class have CM by $\mathcal{O}_K$ (the idea being that testing for the isogeny class is faster than for the endomorphism ring).

Sutherland in [Sut10] suggested to modify the CRT algorithm as follow:

Algorithm 1.2.6 :

```
 INPUT: An imaginary quadratic field $K$, and a collection of CRT primes $P_K$ for $K$.
OUTPUT: The class polynomial $H(x)$ either in $\mathbb{Z}[x]$ or reduced modulo a prime $q$.

1) Loop through CRT primes $p \in P_K$:
    (a) Enumerate elliptic curves $E$ over $\mathbb{F}_p$ until a curve in the right isogeny class is found.
    (b) Use vertical isogenies to find from $E$ a curve with CM by $\mathcal{O}_K$ ("go up in the volcano").
    (c) From a maximal curve $E$, use isogenies to compute all other maximal curves.
    (d) Reconstruct the class polynomial $H(X)$ modulo $p$ from the $j$-invariants of the set of
        maximal curves.
2) Recover $H(X)$ in $\mathbb{Z}[X]$ or modulo $q$ using the (explicit) CRT method once we have
   computed $H(X)$ modulo $p$ for enough primes $p$.                                        ♢
```

By contrast with the preceding method, the new algorithm tries to select CRT primes such that the index $[\mathcal{O}_K : \mathbb{Z}[\pi]]$ is divisible by a lot of small prime factors. The idea behind Algorithm 1.2.6 is to maximise the size of the isogeny class to speed up the search for a curve in this class. Once such a curve is found, going-up in the volcano is pretty fast. Overall, while the new algorithm has the same asymptotic complexity, it gives a huge speed-up in practice, and it is currently the fastest algorithm to compute class polynomials in genus 1.

Often, for quasi-linear algorithms, they are memory-bound rather than CPU-bound. One huge advantage of the CRT method is that by using the explicit CRT, it can compute the class polynomial modulo a CRT prime q of cryptographic size without having to compute it over $\mathbb{Z}$. At such it does not suffer from the same memory problem as the analytic method.

Finally, we mention the article [IJ10] that allows to select which isogeny can go up in the volcano in advance by doing some Tate pairing computations.

1.2.5 Class invariants

Since all the presented algorithms are quasi-linear in the size of the class polynomial H, one can try to replace H by a smaller class polynomial. This can be done by replacing the j-invariant by another class invariant to represent the moduli points of elliptic curves with CM by $\mathcal{O}_K$. For instance if all such curves have a rational level n structure (they correspond to rational points on $X_0(n)$ or $X_1(n)$, which typically happens when n has some splitting behaviour in K), one can use modular coordinates on $X_0(n)$ rather than on $X(1)$.

It is straightforward to adapt the analytic and p-adic method to use these class invariants, but a bit more subtle for the CRT algorithm [ES10a].

1.3 The case of genus 2

An abelian variety $A/\mathbb{C}$ of dimension 2 is said to have complex multiplication if End(A) is an order in K, a CM field of degree 4, that is, an imaginary-quadratic extension of a real-quadratic field K_0. If A is simple (not isogenous to a product of elliptic curves) then K is a primitive CM field. In this case, either K is Galois with cyclic Galois group, or the Galois closure L of K is a CM field of degree 8 with Galois group the dihedral group D_4. (If K is non-primitive as a CM field, then it is Galois with bicyclic Galois group).

Two complications arise here compared to the elliptic curve case. First, when K is non-Galois, there are two non-equivalent CM types (we refer to Chapter 2 for definitions). Indeed, if $A/\mathbb{C}$ has CM by $\mathcal{O}_K$, then the action of End(A) and the tangent space at 0 determine a CM type. Reciprocally, if $\mathfrak{a} \subset \mathcal{O}_K$ is an ideal, the embedding of $\mathfrak{a}$ into a lattice in $\mathbb{C}^2$ depends on the choice of the CM type. As such, the moduli space $\mathcal{M}_{\mathcal{O}_K}$ of all abelian varieties with CM by $\mathcal{O}_K$ is split into two components $\mathcal{M}_{\Phi_1}$ and $\mathcal{M}_{\Phi_2}$ for the two inequivalent CM types Φ_1 and φ_2. Note that while $\mathcal{M}_{\mathcal{O}_K}$ is defined over $\mathbb{Q}$, the components $\mathcal{M}_{\Phi_i}$ are defined over K_0^r, the real subfield of the reflex field K^r for Φ_i. (Note that while the reflex field depends on the CM type Φ, this is not the case for K_0^r.) The Galois action of Gal($K_0^r/\mathbb{Q}$) permutes these two components.

The second difficulty is that while the Neron-Severi group of an elliptic curve is isomorphic to $\mathbb{Z}$, this is no longer the case in dimension 2, where it is acted upon by $\mathcal{O}_{K_0}$, the maximal order of K_0. So the class group describing the abelian extension in which the points of the moduli space $\mathcal{M}_\Phi$ live is more complicated than for the genus 1 case.

The extension of Deuring's result was worked out by Taniyama-Shimura-Weil. First we recall that Igusa gave coordinates for the moduli space of abelian varieties of dimension 2, the so-called Igusa invariants j_1, j_2, j_3. In the following, we will denote $j = (j_1, j_2, j_3)$. We also note that if j is the Igusa invariant of a Jacobian of a hyperelliptic curve of genus 2, then then we can reconstruct the curve using the Mestre-Cardona-Quer [Mes91; CQ05] algorithm.

Define the Shimura class group as

$$\mathfrak{C} = \{(\mathfrak{a}, \rho) \mid \mathfrak{a} \text{ a fractional } \mathcal{O}_K\text{-ideal with } \mathfrak{a}\overline{af} = (\rho),\, \rho \in K_0 \text{ totally positive}\}/K^*$$

If a CM type Φ is fixed, then Φ extends to a CM type on the Galois closure L of K, and Φ^{-1} descend into a CM type on the reflex field K^r (the reflex CM type Φ^r). From this CM type, one can then define the type norm $N_{\Phi^r} : \mathrm{Cl}_{K^r} \to \mathfrak{C}$.

Theorem 1.3.1 (Shimura) : *The moduli space $\mathcal{M}_\Phi$ is a torsor (a principal homogeneous space) under $\mathfrak{C}$ (where $\mathfrak{C}$ acts by isogenies). If $A \in \mathcal{M}_\Phi$, then $\mathfrak{H} = K^r(j(A))$ is an abelian extension of K^r corresponding to the class group $N_{\Phi^r}(\mathrm{Cl}_{K^r})$.*

Moreover, the field $\mathfrak{H}_0 = K_0^r(j(A))$ is linearly disjoint from K^r over K_0^r and we have $\mathfrak{H} = \mathfrak{H}_0 K^r$. The extension $\mathfrak{H}/K_0^r$ is Galois, and $\mathfrak{H}_0$ is the real subfield of the CM field $\mathfrak{H}$.

The moduli space $\mathcal{M}_\Phi$ splits into irreducible components under the action of $\mathrm{Gal}(\mathfrak{H}/K)$. These components correspond to the orbits of the action of $N_{\Phi^r}(\mathrm{Cl}_{K^r})$ in $\mathfrak{C}$, and are defined over K_0^r. The action of $\sigma \in \mathrm{Gal}(K_0^r/\mathbb{Q})$ sends an irreducible component of $\mathcal{M}_\Phi$ to an irreducible component of $\mathcal{M}_{\sigma\Phi}$. If K is cyclic Galois, there is only one CM type, so these irreducible components are defined over $\mathbb{Q}$.

Proof : See [Str10, Theorem I.9.1 and Chapter III]. ■

We refer the reader interested in the theory of CM algebras and the extension by Deligne of Shimura's main theorem of complex multiplication over K_0^r to a theorem over $\mathbb{Q}$ to the online notes by Milne [Mil06].

In practice, one represents an irreducible component of $\mathcal{M}_\Phi$ (which is then of dimension 0) by using a Hecke representation given by three polynomials on the three Igusa invariants.

The three methods to compute class polynomials have been extended to genus 2 for computing the Igusa class polynomials:

1) the complex analytic method in [Spa94; Wam99; Wen03; Str10; ET13];

2) the p-adic lifting method in [GHK+06; CKL08; CL09];

3) the Chinese remainder theorem method (CRT) in [EL10; FL08; BGL11; LR12].

The aim of this report is to give an overview of these methods in genus 2. In Chapter 2 we work out explicit equations for the reflex field, and give algorithms to compute the Shimura class group and the image of the type norm. In that chapter, we also explain how to obtain symbolic period matrices representing abelian variety with CM by $\mathcal{O}_K$.

In Chapter 3 we explain how the abelian varieties with CM by $\mathcal{O}_K$ (and the class polynomials) reduce to a finite field, and how to compute the corresponding Weil polynomial giving their isogeny class (and in particular their number of points). While this could have been part of the previous chapter, we put it in a chapter of its own because if one is given the class polynomials, one only needs to read this chapter to understand how to use these polynomials to generate suitable abelian varieties over a finite field.

Chapter 4 gives an overview of the analytic method. The main idea is to use a fast algorithm based on the AGM to evaluate in quasi-linear time (in the precision) the theta constants associated to the symbolic matrices obtained in Chapter 2. From the theta constants, it is easy to recover the Igusa invariants, and then the Igusa class polynomials as in §1.2.2.

Chapter 5 describes several algorithms related to abelian varieties of dimension 2 over finite fields. We describe how to compute isogenies and endomorphism rings. We also explain how one can use “vertical” isogenies to go from a variety in the right isogeny class to a variety with maximal CM by $\mathcal{O}_K$; and how to use “horizontal” isogenies to go from a variety with CM by $\mathcal{O}_K$ to all varieties with CM by $\mathcal{O}_K$. We explain how these tools are used by the p-adic method in Chapter 6 and by the CRT method in Chapter 7.

We conclude by examples of computations of class polynomials in Chapter 8.

As progress is still being made on this subject, we also mention during the report some work not yet published.

1.3.1 Complexity

In the original articles about computing class polynomials in genus 2, the authors focus on the full class polynomials of $\mathscr{M}_{\mathcal{O}_K}$ defined over $\mathbb{Q}$. In this report, we instead focus on the class polynomials over K_0^r corresponding to an irreducible component of $\mathscr{M}_\Phi$, since it will be smaller. If one is interested in obtaining the full class polynomials over $\mathbb{Q}$, it is easy to extend the methods presented here.

We note that contrarily to the case of genus 1, the class polynomials have denominators in their coefficients (the primes p that appear correspond to primes $\mathfrak{p} \in \mathfrak{H}$ over p such that the reduction of an abelian variety with CM by $\mathcal{O}_K$ modulo $\mathfrak{p}$ is not absolutely simple, i.e. is isogenous to a product of elliptic curves).

The Bruinier-Yang conjectural formulæ [BY06] (proved only for special cases [Yan10a; Yan10b]) explain which prime powers can appear in these denominators. These formulas have been recently improved in [GJL+11; LV12; ABL+12]. We note that we use Igusa invariants from [Str10, Appendix 3] rather than the ones originally used by Spallek in [Spa94]. They have the advantage that the power of h_{10} in the denominators is $1, 2, 2$ rather than $6, 4, 4$; but we have to correct the denominators from the Bruinier-Yang formula by the constant factors $2^7, 2^5, 2^{14}$)].

In practice, it is sufficient to simply do an LLL lift, as explained in §4.3.

Let $\Delta_0 = \Delta_{K_0/\mathbb{Q}}$ and $\Delta_1 = N_{K_0/\mathbb{Q}}(\Delta_{K/K_0})$, so that $\Delta = \Delta_{K/\mathbb{Q}} = \Delta_1 \Delta_0^2$. Then the degree of the class polynomials is $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$, while the height of their coefficients is bounded by $\widetilde{O}(\Delta_0^{5/2}\Delta_1^{3/2})$ (see [Str10, §II.9] and [GL12]). In practice, we observe [Str10, Appendix 3] that the height of the coefficient seems to be bounded by $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$, and we will use this observed bound in the following analysis.

In total, the size of the class polynomials is thus $\widetilde{O}(\sqrt{\Delta_0 \Delta_1})$. Then we have the following complexities for the three methods. The analytic method is quasi-linear in the size of the class polynomials. The p-adic method as published is quasi-cubic, but we argue in Chapter 6 that with the use of isogenies (as presented in Chapter 5), it should be possible to obtain an algorithm of complexity $\widetilde{O}((\Delta_0\Delta_1)^{5/2})$. We then give a potential method to achieve $\widetilde{O}((\Delta_0\Delta_1)^{3/2})$ or perhaps even quasi-linear time (by using real multiplication).

Unfortunately, the CRT has for now a quasi-cubic complexity. This is in part due to the fact that we do not have yet general enough isogenies in order to be always able to go up. But even if this were the case, the size of the isogeny class is too small compared to the set of all hyperelliptic curves of genus 2. Likewise, we give a potential method to obtain a quasi-quadratic complexity or perhaps even a complexity in $\widetilde{O}((\Delta_0\Delta_1)^{3/2})$.

As in the genus 1 case, one could also look for smaller class polynomials by using other class invariants. We refer for this to [Str] and to as of yet unpublished work by Enge and Streng generalising [Sch02] to the genus 2 case.

We also mention the problem of checking the correctness of the computations. A problem is that the proved bound on the height of the coefficients is a lot larger than the real height of the class polynomials, and moreover we do not have explicit bounds. In practice the analytic and p-adic methods stop as soon as they have enough precision to be able to recover algebraic coefficients. The CRT method stops as soon as an extra prime give the same polynomial with the CRT reconstruction. In a sense the correctness of the CRT method is a bit better behaved: it will be wrong if all the differences between the coefficients of the real class polynomials compared to the computed one are divisible by the current prime (which is unlikely to happen, see [FL08, Remark 7.2]).

Still, for cryptographic applications, one can check that the curve obtained by reducing the class polynomials modulo a prime of cryptographic size has the expected number of points (which is much easier to check than doing a point counting algorithm). One could even check that it has the right endomorphism ring using the results from Chapter 5. Finally, a quick heuristic check of the class polynomials can be done by checking if they have the correct splitting behaviour when reduced modulo several primes according to the theory of Chapter 3.

2 CM theory for genus 2

2.1 Complex multiplication theory

In this section, we provide a concise introduction to the theory of complex multiplication of principally polarised abelian surfaces or, equivalently, Jacobians of genus 2 hyperelliptic curves over the complex numbers, to the extent needed to describe the algorithms and implementation. The presentation follows [Str10], and proofs are given in [ST61; Shi98; Str10; Str09].

2.1.1 Quartic CM fields and abelian surfaces

A *CM field* K is an imaginary-quadratic extension of a totally real number field K_0. We denote by $\varkappa$ indiscriminately the complex conjugation on $\mathbb{C}$ and the automorphism generating $\mathrm{Gal}(K/K_0)$. For any embedding $\varphi : K \to \mathbb{C}$, we have $\varkappa \circ \varphi = \varphi \circ \varkappa$, which justifies the notation $\overline{\varphi} = \varkappa \circ \varphi$.

Quartic CM fields K of degree 4 over $\mathbb{Q}$ come in three Galois types. Generically, $K/\mathbb{Q}$ is not Galois, the Galois closure L/K is of degree 2, and $\mathrm{Gal}(L/\mathbb{Q})$ is isomorphic to the dihedral group D_4. L is itself a CM field, and the complex conjugation of L, which we denote again by $\varkappa$, restricts to the complex conjugation of K. If $K/\mathbb{Q}$ is Galois, it may be either cyclic or biquadratic. We will not consider the biquadratic case in the following, since then the abelian surfaces of which it is the endomorphism algebra are products of elliptic curves; so from now on, all Galois quartic CM fields are tacitly understood to be cyclic.

A *CM type* of a quartic CM field K is a set $\Phi = \{\varphi_1, \varphi_2\}$ of two embeddings $K \to \mathbb{C}$ such that $\varphi_2 \neq \overline{\varphi}_1$; that is, it contains one out of each pair of complex-conjugate embeddings. Two CM types Φ and Φ' are equivalent if there is an automorphism σ of K such that $\Phi' = \Phi \circ \sigma$; in particular, Φ and $\overline{\Phi}$ are equivalent. If $K/\mathbb{Q}$ is Galois, there is only one equivalence class of CM types; otherwise, there are two inequivalent classes $\Phi = \{\varphi_1, \varphi_2\}$ and $\Phi' = \{\varphi_1, \overline{\varphi}_2\}$.

For a given CM type $\Phi = \{\varphi_1, \varphi_2\}$, its *reflex field* is the field K^r generated over $\mathbb{Q}$ by the *type traces*, that is, $K^r = \mathbb{Q}\big(\{\varphi_1(x) + \varphi_2(x) : x \in K\}\big)$; it is itself a quartic CM field and we denote by K_0^r its real-quadratic subfield. Equivalent CM types yield conjugate reflex fields. In the Galois case, K and K^r are isomorphic, while in the dihedral case, they are not isomorphic, but the two reflex fields for the two inequivalent CM types are. In both cases, there is a natural way of defining a dual CM type $\Phi^r = \{\varphi_1^r, \varphi_2^r\}$ of K^r, and the reflex field of K^r is isomorphic to K. Define the (dual) *type norm* $\mathrm{N}_{\Phi^r} : K^r \to K$ by $x \mapsto \varphi_1^r(x)\varphi_2^r(x)$, so that

$$\mathrm{N}_{\Phi^r}\, \overline{\mathrm{N}}_{\Phi^r} = \mathrm{N}; \tag{2.1}$$

this map extends to ideals and ideal classes.

In §2.2, we provide explicit equations for all occurring number fields and consider their embeddings from an effective point of view.

Let $\mathfrak{a}$ be a fractional ideal of $\mathcal{O}_K$. A CM type $\Phi = \{\varphi_1, \varphi_2\}$ induces an embedding $K \to \mathbb{C}^2$, $x \mapsto (\varphi_1(x), \varphi_2(x))$, under which $\Phi(\mathfrak{a})$ is a lattice of rank 4. Its cokernel $\mathbb{C}^2/\Phi(\mathfrak{a})$, a complex torus of genus 2, is an *abelian surface*. Let $\mathcal{\delta}_K^{-1} = \{y \in K : \mathrm{Tr}(xy) \in \mathbb{Z}\ \forall x \in \mathcal{O}_K\}$ be the codifferent ideal of K. Assume that $(\mathfrak{a}\overline{\mathfrak{a}}\mathcal{\delta}_K)^{-1}$ is principal and generated by some $\xi \in K$ such that $\varphi_1(\xi), \varphi_2(\xi) \in i\mathbb{R}^{>0}$; in particular, $\xi\overline{\xi} \in K_0$ is totally negative. Then $E_{\Phi,\xi} : \Phi(K)^2 \to \mathbb{Q}, (\Phi(x), \Phi(y)) \mapsto \mathrm{Tr}(\xi\overline{x}y)$ is a symplectic form over $\mathbb{Q}$ which takes integral values on $\Phi(\mathfrak{a})^2$. By tensoring with $\mathbb{R}$, one obtains a symplectic form $\mathbb{C}^2 \to \mathbb{R}$ such that $(x, y) \mapsto E_{\Phi,\xi}(ix, y)$ is symmetric and positive definite, a *principal polarisation* on $\mathbb{C}^2/\Phi(\mathfrak{a})$.

The principally polarised abelian surface $A(\Phi, \mathfrak{a}, \xi) = \left(\mathbb{C}^2/\Phi(\mathfrak{a}), E_{\Phi,\xi}\right)$ has complex multiplication by $\mathcal{O}_K$; conversely, any such surface can be obtained up to isomorphism in this way. Two principally polarised abelian surfaces $A(\Phi, \mathfrak{a}, \xi)$ and $A(\Phi', \mathfrak{a}', \xi')$ are isomorphic if and only if $\Phi = \Phi'$ (up to equivalence) and there is a $u \in K^*$ such that $\mathfrak{a}' = u\mathfrak{a}$ and $\xi' = (u\overline{u})^{-1}\xi$. In particular this implies that $u\overline{u} \in K_0$ is totally positive, and that we may assume $\mathfrak{a}$ to be an integral ideal of $\mathcal{O}_K$.

2.1.2 The Shimura group, its type norm subgroup and cosets

The Igusa invariants to be defined in §2.1.3 determine the moduli space $\mathcal{M}$ of principally polarised complex abelian surfaces, which has a model over $\mathbb{Q}$. Let $\mathcal{M}_{K,\Phi}$ be the subset of surfaces $A(\Phi, \mathfrak{a}, \xi)$ obtained from an integral ideal of $\mathcal{O}_K$ and the CM type Φ as described in §2.1.1. Then $\mathcal{M}_{K,\Phi}$ is stable under $\mathrm{Gal}(\overline{\mathbb{Q}}/K_0^r)$. If K is cyclic, then $\mathcal{M}_{K,\Phi}$ is even stable under $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$. Otherwise let Φ' be inequivalent with Φ. Then $\mathcal{M}_{K,\Phi}$ and $\mathcal{M}_{K,\Phi'}$ are disjoint and conjugate under $\mathrm{Gal}(K_0^r/\mathbb{Q})$ [Str10, Lemmata 1.1 and 2.1].

Let the *Shimura class group* $\mathfrak{C}$ be defined by

$$\mathfrak{C} = \left\{(\mathfrak{a}, u) : \mathfrak{a} \text{ a fractional ideal of } \mathcal{O}_K, \mathfrak{a}\overline{\mathfrak{a}} = u\mathcal{O}_K, \text{ and } u \in K_0 \text{ totally positive}\right\}/\sim \tag{2.2}$$

with component-wise multiplication. The equivalence relation denoted $\sim$ above is the one induced by principal ideals, more precisely the equivalence modulo the subgroup given by the $(v\mathcal{O}_K, v\overline{v})$ with $v \in K^*$ and $v\overline{v} \in K_0$ totally positive.

By the discussion of §2.1.1, $\mathfrak{C}$ acts regularly on $\mathcal{M}_{K,\Phi}$ via

$$(\mathfrak{b}, u) \cdot A(\Phi, \mathfrak{a}, \xi) = A(\Phi, \mathfrak{b}^{-1}\mathfrak{a}, u\xi). \tag{2.3}$$

Consider the dual type norm map $\mathrm{N}_{\Phi^r} : \mathrm{Cl}_{K^r} \to \mathfrak{C}, \mathfrak{b} \mapsto (\mathrm{N}_{\Phi^r}(\mathfrak{b}), \mathrm{N}(\mathfrak{b}))$, which is well-defined by (2.1). For any $A(\Phi, \mathfrak{a}, \xi)$, the action induced by $\mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})$ is that of the Galois group of the field of moduli of $A(\Phi, \mathfrak{a}, \xi)$ over K^r [Str10, Theorem 9.1]; otherwise said, the field of moduli is the fixed field of $\ker(\mathrm{N}_{\Phi^r})$ inside the Hilbert class field of K^r. The cokernel of N_{Φ^r} is elementary abelian of exponent 1 or 2 [Str10, Theorem 2.2], so $\mathcal{M}_{K,\Phi}$ splits into orbits under $\mathfrak{C}$ of size $|\mathrm{im}(\mathrm{N}_{\Phi^r})|$, and the number of orbits is a power of 2. As stated above, these orbits are in fact defined over K_0^r, with the orbits of $\mathcal{M}_{K,\Phi}$ and $\mathcal{M}_{K,\Phi'}$ being mapped to each other by $\mathrm{Gal}(K_0^r/\mathbb{Q})$.

2.1.3 ϑ-functions, Igusa invariants and class polynomials

Given an ideal $\mathfrak{a}$ and a principal polarisation $E_{\Phi,\xi}$ as in §2.1.1, one may choose a $\mathbb{Z}$-basis $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ of $\mathfrak{a}$ such that $v_1 = \Phi(\alpha_1)$, $v_2 = \Phi(\alpha_2)$, $w_1 = \Phi(\alpha_3)$, $w_2 = \Phi(\alpha_4)$ form a symplectic basis, for which $E_{\Phi,\xi}$ becomes $\begin{pmatrix} 0 & \mathrm{id}_2 \\ -\mathrm{id}_2 & 0 \end{pmatrix}$. That the change of basis is defined over $\mathbb{Z}$ and not only over $\mathbb{R}$ follows from the principality of the polarisation; we also call this basis of $\mathfrak{a}$ *symplectic*. Let $V = \begin{pmatrix} v_1 & v_2 \end{pmatrix}$, $W = \begin{pmatrix} w_1 & w_2 \end{pmatrix} \in \mathbb{C}^{2\times 2}$. Rewriting the ambient vector space $\mathbb{C}^2$ and $\Phi(\mathfrak{a})$ in the basis spanned by w_1 and w_2, we obtain $\Phi(\mathfrak{a}) = \begin{pmatrix} \Omega_{\Phi,\mathfrak{a},\xi} & \mathrm{id}_2 \end{pmatrix} \mathbb{Z}^4$ with the *period matrix*

$$\Omega_{\Phi,\mathfrak{a},\xi} = W^{-1}V \tag{2.4}$$

in the *Siegel half space* $\mathscr{H}_2 = \{\Omega \in \mathbb{C}^{2\times 2} : \Omega \text{ symmetric and } \Im(\Omega) \text{ positive definite}\}$. The symplectic group $\mathrm{Sp}_4(\mathbb{Z})$ acts on $\mathscr{H}_2$ by

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \Omega = (A\Omega + B)(C\Omega + D)^{-1},$$

where $A, B, C, D \in \mathbb{Z}^{2\times 2}$. As in the case of genus 1, a fundamental domain for $\mathscr{H}_2$ exists under the action of $\mathrm{Sp}_4(\mathbb{Z})$. Reduction into the fundamental domain is discussed in §2.2.3.

ϑ-constants are certain modular forms of weight 1/2 for $\mathrm{Sp}_4(\mathbb{Z})$. Let $a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$, $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \in \left(\frac{1}{2}\mathbb{Z}\right)^2$ be two vectors of *ϑ-characteristics*. Then for $\Omega \in \mathscr{H}_2$,

$$\vartheta_{16a_1+8a_2+4b_1+2b_2}(\Omega) = \vartheta_{a,b}(\Omega) = \sum_{n\in\mathbb{Z}^2} e^{2\pi i\left(\frac{1}{2}(n+a)^\intercal \Omega (n+a) + (n+a)^\intercal b\right)}. \tag{2.5}$$

Only the *even* ϑ-constants ϑ_i for $i \in T = \{0,1,2,3,4,6,8,9,12,15\}$ are not identically 0.

The following *duplication formulæ* relate the values of the squares of the ten even ϑ-constants in the argument Ω with the values of the four *fundamental ϑ-constants* $\vartheta_0, \ldots, \vartheta_3$ (which have $a = 0$) in the argument $\Omega/2$ (omitted from the formulæ for the sake of conciseness).

$$\begin{aligned}
4\vartheta_0^2(\Omega) &= \vartheta_0^2 + \vartheta_1^2 + \vartheta_2^2 + \vartheta_3^2 & 4\vartheta_6^2(\Omega) &= 2\vartheta_0\vartheta_2 - 2\vartheta_1\vartheta_3 \\
4\vartheta_1^2(\Omega) &= 2\vartheta_0\vartheta_1 + 2\vartheta_2\vartheta_3 & 4\vartheta_8^2(\Omega) &= \vartheta_0^2 + \vartheta_1^2 - \vartheta_2^2 - \vartheta_3^2 \\
4\vartheta_2^2(\Omega) &= 2\vartheta_0\vartheta_2 + 2\vartheta_1\vartheta_3 & 4\vartheta_9^2(\Omega) &= 2\vartheta_0\vartheta_1 - 2\vartheta_2\vartheta_3 \\
4\vartheta_3^2(\Omega) &= 2\vartheta_0\vartheta_3 + 2\vartheta_1\vartheta_2 & 4\vartheta_{12}^2(\Omega) &= \vartheta_0^2 - \vartheta_1^2 - \vartheta_2^2 + \vartheta_3^2 \\
4\vartheta_4^2(\Omega) &= \vartheta_0^2 - \vartheta_1^2 + \vartheta_2^2 - \vartheta_3^2 & 4\vartheta_{15}^2(\Omega) &= 2\vartheta_0\vartheta_3 - 2\vartheta_1\vartheta_2
\end{aligned} \tag{2.6}$$

Denote by h_j the following modular forms of weight j:

$$\begin{aligned}
h_4 &= \sum_{i\in T} \vartheta_i^8, & h_6 &= \sum_{15 \text{ triples } (i,j,k)\in T^3} \pm(\vartheta_i\vartheta_j\vartheta_k)^4, \\
h_{10} &= \prod_{i\in T} \vartheta_i^2, & h_{12} &= \sum_{15 \text{ tuples } (i,j,k,l,m,n)\in T^6} (\vartheta_i\vartheta_j\vartheta_k\vartheta_l\vartheta_m\vartheta_n)^4;
\end{aligned} \tag{2.7}$$

for the exact definitions, see [Str10, §II.7.1]. These generate the ring of holomorphic Siegel modular forms over $\mathbb{C}$, see [Igu62, Corollary p. 195] and [Str10, Remark 7.2]. The moduli space of principally polarised abelian surfaces is of dimension 3 and parameterised by *absolute Igusa invariants*, modular functions (thus of weight 0) in $\mathbb{Z}\left[h_4, h_6, h_{12}, h_{10}^{-1}\right]$. Different sets of invariants have been suggested in the literature. The most cited one is Spallek's, who uses a system in the linear span of $\frac{h_{12}^5}{h_{10}^6}, \frac{h_{12}^3 h_4}{h_{10}^4}, \frac{h_{12}^2 h_6}{h_{10}^3}$ [Spa94, Satz 5.2]. Streng defines invariants with the minimal powers of h_{10} in the denominator as

$$j_1 = \frac{h_4 h_6}{h_{10}}, \quad j_2 = \frac{h_4^2 h_{12}}{h_{10}^2}, \quad j_3 = \frac{h_4^5}{h_{10}^2}. \tag{2.8}$$

The principally polarised abelian surfaces $A(\Phi, \mathfrak{a}, \xi)$ are parameterised by the triples of *singular values* $(j_1(\Omega), j_2(\Omega), j_3(\Omega))$ in the period matrices $\Omega = \Omega_{\Phi,\mathfrak{a},\xi}$, which may be obtained from the action of the Shimura class group $\mathfrak{C}$ on a fixed *base point* $\beta = (\Phi, \mathfrak{a}_\Phi, \xi_\Phi)$. The singular values lie in the subfield of the Hilbert class field of K^r given in §2.1.2. Following the discussion there, the *Igusa class polynomials* $I_i(X) = \prod_{(\Phi,\mathfrak{a},\xi)} \left(X - j_i(\Omega_{\Phi,\mathfrak{a},\xi})\right)$ are defined over $\mathbb{Q}$. More precisely their irreducible factors, over K_0^r in the dihedral case or $\mathbb{Q}$ in the cyclic case, are given by

$$\prod_{C \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})} \left(X - j_i(\Omega_{CC'\cdot\beta})\right),$$

where Φ is one CM type and $C' \in \mathfrak{C}/\mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})$.

In the following, we fix a CM type Φ (for its explicit description, see §2.2) and a base point $\beta = (\Phi, \mathfrak{a}_\Phi, \xi_\Phi)$ and let

$$H_1(X) = \prod_{C \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})} \left(X - j_1(\Omega_{C\cdot\beta})\right). \tag{2.9}$$

As elements of the same class field, the singular values of j_2 and j_3 are rational expressions in the singular value of j_1. Computationally, it is preferable to use the *Hecke representation* in the trace-dual basis to keep denominators small. We thus define polynomials $\widehat{H}_2$ and $\widehat{H}_3$ through $j_i H_1'(j_1) = \widehat{H}_i(j_1)$ with

$$\widehat{H}_i(X) = \sum_{C \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})} j_i(\Omega_{C\cdot\beta}) \prod_{D \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) \setminus \{C\}} \left(X - j_1(\Omega_{D\cdot\beta})\right) \tag{2.10}$$

for $i \in \{2,3\}$, where H_1, $\widehat{H}_2$, $\widehat{H}_3 \in K_0^r[X]$ in the dihedral case and $\in \mathbb{Q}[X]$ in the cyclic case.

2.2 Explicit equations and symbolic period matrices

While Algorithm 4.1.1 *in fine* works with complex approximations obtained *via* CM types, it starts from an algebraic setting. In this section, we examine how to carry out the computations as far as possible symbolically with algebraic numbers, which relieves us from the need to decide on the necessary precision early on. In particular, in §2.2.1 we replace the complex embeddings forming a CM type by algebraic embeddings into the compositum L of all involved fields, followed by a "universal" embedding ψ of L into $\mathbb{C}$. Taking preimages under ψ, the entries of the period matrices $\Omega \in \mathbb{C}^{2\times 2}$ may then be interpreted as elements of the reflex field and may be handled symbolically. We then fix a model for the CM field K in §2.2.2 and derive explicit equations for all considered fields and embeddings.

Recall the notation of §2.1: K is a quartic CM field, K_0 its real quadratic subfield and L its Galois closure with Galois group G. We consider only the dihedral case $[L:K] = 2$ and $G = D_4$ and the cyclic case $L = K$ and $G = C_4$. Let $\Phi = (\varphi_1, \varphi_2)$ be a CM type, where φ_1, $\varphi_2 : K \to \mathbb{C}$ are two complex embeddings of K with $\varphi_2 \neq \overline{\varphi}_1$, and let K^r be the reflex field of K with respect to Φ.

2.2.1 Galois theory, embeddings and period matrices

The dihedral case

Galois theory. We have the following diagram of fields and Galois groups:

$$\begin{tikzcd}
 & & L=KK^r \arrow[dll, dash, "\langle\rho\rangle"'] \arrow[d, dash, "\langle\varkappa\rangle"] \arrow[drr, dash, "\langle\sigma\rangle"] & & \\
K \arrow[d, dash, "\langle\varkappa|_K\rangle"'] & & * \arrow[dll, dash] \arrow[drr, dash] & & K^r \arrow[d, dash, "\langle\varkappa|_{K^r}\rangle"] \\
K_0 \arrow[drr, dash] & & & & K_0^r \arrow[dll, dash] \\
 & & \mathbb{Q} & &
\end{tikzcd}$$

As an abstract group, the dihedral group D_4 is generated by two elements τ, ρ with the relations $\tau^4=1$, $\rho^2=1$ and $\rho\tau\rho=\tau^3$. It contains two elements of order 4, τ and τ^3, all other non-unit elements are of order 2. Its centre is generated by $\tau^2=(\tau^3)^2$.

Let $\mathrm{Gal}(L/K)=\langle\rho\rangle$ and $\mathrm{Gal}(L/K^r)=\langle\sigma\rangle$. Let $\varkappa\in G$ be complex conjugation; by [Str10, Lemma I.2.2(2)] it lies in the centre of G and is thus the square of the elements of order 4. We have $\mathrm{Gal}(L/K_0)=\langle\rho,\varkappa\rangle$ and $\mathrm{Gal}(L/K_0^r)=\langle\sigma,\varkappa\rangle$. Since $K_0\neq K_0^r$, their intersection equals $\mathbb{Q}$, so $G=\mathrm{Gal}(L/\mathbb{Q})=\langle\rho,\sigma,\varkappa\rangle$. As G is not commutative, but ρ and σ commute with $\varkappa$, we have $\rho\sigma\neq\sigma\rho$; from ρ and σ being of order 2 we then deduce that $\tau=\rho\sigma$ is of order 4, so that $\varkappa=\tau^2$. We may then consider τ and ρ as the dihedral generators of G.

Embeddings and CM types. There is a unique embedding $\psi: L\to\mathbb{C}$ such that $\varphi_1=\psi|_K$ and $\varphi_2=(\psi\sigma)|_K$ (where multiplication denotes composition), which can be seen as follows. First of all, there are two embeddings which, restricted to K, yield φ_1; we denote them by ψ_1 and $\psi_1'=\psi_1\rho$. Now there is $s\in G$, uniquely defined up to multiplication by ρ from the right, such that $\varphi_2=(\psi_1 s)|_K$. Since $\varphi_2\neq\varphi_1$ and $\varphi_2\neq\overline{\varphi}_1$, the automorphism s is neither 1, ρ, $\varkappa=\tau^2$ nor $\varkappa\rho=\tau^2\rho$. This leaves s as one of $\tau=\rho\sigma$, $\tau\rho=\rho\sigma\rho$, $\tau^3=\sigma\rho$ or $\tau^3\rho=\sigma$. If $s|_K=\sigma|_K=(\sigma\rho)|_K$, we may choose $\psi=\psi_1$. Otherwise, $s|_K=\rho\sigma$, and $(\psi_1'\sigma)|_K=(\psi_1\rho\sigma)|_K=(\psi_1 s)|_K=\varphi_2$, so we choose $\psi=\psi_1'$.

Period matrices. Let $(\alpha_1,\ldots,\alpha_4)$ be a symplectic basis for the ideal $\mathfrak{a}$ of K with respect to $E_{\Phi,\xi}$ as defined in §2.1.3. Then

$$\begin{aligned}
V &= \begin{pmatrix}\varphi_1(\alpha_1) & \varphi_1(\alpha_2)\\ \varphi_2(\alpha_1) & \varphi_2(\alpha_2)\end{pmatrix}=\psi\left(\begin{pmatrix}\alpha_1 & \alpha_2\\ \alpha_1^\sigma & \alpha_2^\sigma\end{pmatrix}\right),\\
W &= \begin{pmatrix}\varphi_1(\alpha_3) & \varphi_1(\alpha_4)\\ \varphi_2(\alpha_3) & \varphi_2(\alpha_4)\end{pmatrix}=\psi\left(\begin{pmatrix}\alpha_3 & \alpha_4\\ \alpha_3^\sigma & \alpha_4^\sigma\end{pmatrix}\right)
\end{aligned}$$

and

$$\Omega_{\Phi,\mathfrak{a},\xi}=W^{-1}V=\psi(M)\text{ with }M=\frac{1}{\alpha_3\alpha_4^\sigma-\alpha_4\alpha_3^\sigma}\begin{pmatrix}\alpha_4\alpha_1^\sigma-\alpha_1\alpha_4^\sigma & \alpha_4\alpha_2^\sigma-\alpha_2\alpha_4^\sigma\\ \alpha_3\alpha_1^\sigma-\alpha_1\alpha_3^\sigma & \alpha_3\alpha_2^\sigma-\alpha_3\alpha_2^\sigma\end{pmatrix} \tag{2.11}$$

by (2.4). The entries of M are invariant under σ and thus elements of K^r.

Remark 2.2.1 : It is crucial to choose out of the two embeddings $\psi: L\to\mathbb{C}$ that extend φ_1 the one compatible with φ_2. The other one corresponds to the second CM type $\Phi'=(\varphi_1,\overline{\varphi}_2)$ with reflex field $(K^r)'$ and $\mathrm{Gal}\left(L/(K^r)'\right)=\langle\varkappa\sigma\rangle=\langle\rho\sigma\rho\rangle$. ◇

The cyclic case

Here we have the much simpler situation

$$\begin{tikzcd}
K \arrow[d, dash, "\langle \varkappa \rangle = \langle \sigma^2 \rangle"] \\
K_0 \arrow[d, dash] \\
\mathbb{Q}
\end{tikzcd}$$

We may choose $\psi = \varphi_1$. Then there is a uniquely determined $\sigma \in \operatorname{Gal}(K/\mathbb{Q})$ such that $\varphi_2 = \varphi_1 \sigma$, and trivially M of (2.11) has entries in K^r. In general, they will not lie in a subfield: Since σ is neither the identity nor complex conjugation, it is of order 4.

2.2.2 Number field computations

In this section we show how to express the elements of the reflex field K^r and the normal closure L in consistent ways, so as to be able to compute type norms and entries of period matrices as given by (2.4).

The dihedral case

We use here the same notation for elements of the Galois group G of $L/\mathbb{Q}$ as in §2.2.1.

Field equations. By choosing suitable generating elements we may assume:

$$\begin{aligned} K_0 &= \mathbb{Q}(z) = \mathbb{Q}[Z]/\left(Z^2 + AZ + B\right) \text{ with } A, B > 0, A^2 - 4B > 0; \\ K &= \mathbb{Q}(y) = \mathbb{Q}[Y]/\left(Y^4 + AY^2 + B\right). \end{aligned}$$

We then select the CM type $\Phi = (\varphi_1, \varphi_2)$ with

$$\varphi_1(y) = i\sqrt{\frac{A + \sqrt{A^2 - 4B}}{2}}, \quad \varphi_2(y) = i\sqrt{\frac{A - \sqrt{A^2 - 4B}}{2}}, \tag{2.12}$$

where all the real roots are taken to be positive; the other CM type is $\Phi' = (\varphi_1, \overline{\varphi}_2)$ with $\overline{\varphi}_2(y) = -\varphi_2(y)$. Recall from §2.2.1 the notations $\operatorname{Gal}(L/K) = \langle \rho \rangle$, $\operatorname{Gal}(L/K^r) = \langle \sigma \rangle$, and let $\psi : L \to \mathbb{C}$ be such that $\varphi_1 = \psi|_K$ and $\varphi_2 = (\psi\sigma)|_K$. The reflex field K^r is generated by the type traces of K; letting $y^r = y + y^\sigma$, the equality

$$\psi(y^r) = \psi(y) + (\psi\sigma)(y) = \varphi_1(y) + \varphi_2(y) \tag{2.13}$$

shows that we may consider y^r as a generator of K^r. This gives the equations

$$\begin{aligned} K_0^r &= \mathbb{Q}(z^r) = \mathbb{Q}[Z^r]/\left((Z^r)^2 + A^r Z^r + B^r\right) \text{ with } A^r = 2A, B^r = A^2 - 4B; \\ K^r &= \mathbb{Q}(y^r) = \mathbb{Q}[Y^r]/\left((Y^r)^4 + A^r (Y^r)^2 + B^r\right). \end{aligned}$$

The minimal polynomials of y^r over K and y over K^r follow:

$$(y^r)^2 - 2yy^r + (2y^2 + A), \qquad (y)^2 - y^r y + ((y^r)^2 + A)/2.$$

We write the Galois closure $L = KK^r$ as the compositum generated by $t = y + y^r$. The minimal polynomial of t is the resultant

$$\begin{aligned} h(T) &= \mathrm{Res}_Y\left(Y^4 + AY^2 + B, (T-Y)^2 - 2Y(T-Y) + (2Y^2+A)\right) \\ &= \mathrm{Res}_{Y^r}\left((Y^r)^4 + A^r(Y^r)^2 + B^r, (T-y^r)^2 - y^r(T-y^r) + ((y^r)^2+A)/2\right) \\ &= T^8 + 10AT^6 + (33A^2 - 14B)T^4 + (40A^3 - 70AB)T^2 + 16A^4 - 200A^2B + 625B^2. \end{aligned}$$

Conversions and Galois actions. We are interested in the action of ρ, the generator of $\mathrm{Gal}(L/K)$, on K^r, and in the action of σ, the generator of $\mathrm{Gal}(L/K^r)$, on K. The defining equations give:

$$\begin{gathered} y^r + (y^r)^\rho = 2y, \quad y^r(y^r)^\rho = y^r(y^r)^\rho = 2y^2 + A, \quad y^\rho = y, \\ y + y^\sigma = y^r, \quad yy^\sigma = \left((y^r)^2 + A\right)/2, \quad (y^r)^\sigma = y^r. \end{gathered}$$

An element of K is converted to an element of L, as a relative extension of K^r, using the identity $y = t - y^r$; in the opposite direction we use $y^r = t - y$. The entries of the matrix M of (2.11) are obtained from elements of K and their images under σ, and need to be expressed as elements of K^r. For this we use the identity $y^\sigma = y^r - y$. This allows to work in the relative extension L/K^r and to easily identify elements of K^r.

Dual type norms. For an ideal $\mathfrak{b}$ of K^r, we have

$$\mathrm{N}_{\Phi^r}(\mathfrak{b}) = \mathrm{N}_{L/K}(\mathfrak{b}\mathcal{O}_L),$$

see [BGL11, §3.1]. Computing dual type norms thus reduces to conversions in relative extensions as described above.

The cyclic case

We may use the same type of equations for K and K_0 as in the dihedral case, and may fix $\psi = \varphi_1$ as in (2.12). Fixing an arbitrary element $\sigma \in \mathrm{Gal}(K/\mathbb{Q})$ of order 4, we obtain $\varphi_2 = \varphi_1\sigma$. Then the dual type norm for an ideal $\mathfrak{b}$ of K is computed as

$$\mathrm{N}_{\Phi^r}(\mathfrak{b}) = \mathfrak{b}\overline{\mathfrak{b}}^\sigma,$$

see [BGL11, §3.1].

2.2.3 Symbolic reduction of period matrices

Gottschling in [Got59] has determined a finite set of inequalities describing a fundamental domain $\mathcal{F}_2$ for $\mathrm{Sp}_4(\mathbb{Z})\backslash\mathcal{H}_2$, which directly translate into an algorithm for *reducing* an element of $\mathcal{H}_2$ into $\mathcal{F}_2$. As the Igusa functions introduced in §2.1.3 are modular for $\mathrm{Sp}_4(\mathbb{Z})$, we may transform all period matrices occurring in Algorithm 4.1.1 into $\mathcal{F}_2$. A period matrix Ω is *reduced* if $\mathfrak{R}(\Omega)$ has coefficients between $-\frac{1}{2}$ and $\frac{1}{2}$ (which may be obtained by reducing modulo $\mathbb{Z}$), if the binary quadratic form defined by $\mathfrak{I}(\Omega)$ is reduced (which may be obtained using Gauß's algorithm) and if $|\det(C\Omega + D)| \geqslant 1$ for each of 19 matrices $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathrm{Sp}_4(\mathbb{Z})$ (which may be obtained by applying to Ω a matrix for which the condition is violated). The process needs to be iterated and terminates eventually.

In the light of (2.11), $\Omega = \psi(M)$ with $M \in (K^r)^{2\times 2}$ and an explicitly given $\psi : K^r \to \mathbb{C}$, see (2.13) and (2.12). Letting $K^r = K_0^r + y^r K_0^r$ as before, we have $\psi|_{K_0^r} : K_0^r = \mathbb{Q}(\sqrt{D^r}) \to \mathbb{R}$ and $\psi|_{y^r K_0^r} : y^r K_0^r \to i\mathbb{R}$. So $\Re(M)$ and $\Im(M)$ are the images under ψ of matrices with entries in K_0^r. The condition $|\det(C\Omega+D)| \geqslant 1$ can be rewritten as $\sqrt{\det(C\psi(M)+D)\det(C\psi(\overline{M})+D)} \geqslant 1$ and thus also depends only on the images under ψ of elements of K_0^r.

Hence the period matrices may be transformed symbolically into the fundamental domain $\mathscr{F}_2$ without computing complex approximations of their entries, which precludes rounding errors: The test whether the matrix is reduced and, if not, the decision which transformation to apply depend on the sign of $\psi(\alpha)$ for some $\alpha \in K_0^r$, that is, on the sign of some explicitly known $a + b\sqrt{D^r} \in \mathbb{R}$, where $\sqrt{D^r}$ is the positive root of D^r and $a, b \in \mathbb{Q}$. This sign can be determined from the signs of a and b and the relative magnitudes of a^2 and $b^2 D^r$.

2.3 Computing the Shimura group and its type norm subgroup

2.3.1 Structure of the Shimura group $\mathfrak{C}$

The first step of Algorithm 4.1.1 requires to enumerate the Shimura group $\mathfrak{C}$ of (2.2), or more precisely, its type norm subgroup $\mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})$. We need the following exact sequence, a proof of which can be found in [BGL11]:

$$1 \longrightarrow \mathscr{O}_{K_0}^{+}/N_{K/K_0}(\mathscr{O}_K^{*}) \xrightarrow{u\mapsto(\mathscr{O}_K,u)} \mathfrak{C} \xrightarrow{(\mathfrak{a},\alpha)\mapsto\mathfrak{a}} \mathrm{Cl}_K \overset{N_{K/K_0}}{\dashrightarrow} \mathrm{Cl}_{K_0}^{+} \longrightarrow 1, \tag{2.14}$$

where $\mathscr{O}_{K_0}^{+}$ is the subgroup of totally positive units in $\mathscr{O}_{K_0}$ and $\mathrm{Cl}_{K_0}^{+}$ is the narrow class group of K_0.

We have algorithms at hand for the basic arithmetic of $\mathfrak{C}$. For a finite abelian group, decomposed as a direct product of cyclic groups G_i of order d_i with $d_i \mid d_{i+1}$, we call the d_i the *elementary divisors* and a system of generators of the G_i a *(cyclic) basis* of the group. Such a basis can be computed for the class group Cl_K (quickly under GRH) using the function `bnfinit` in PARI/GP. Equality testing of $(\mathfrak{a},\alpha)$ and $(\mathfrak{b},\beta)$ amounts to testing whether $\mathfrak{a}\mathfrak{b}^{-1}$ is principal (either using `bnfisprincipal` in PARI/GP, or by a direct comparison if each ideal is stored together with its *generalised discrete logarithm*, its coefficient vector with respect to the basis of the class group), and whether $\alpha/\beta = 1$ in $\mathscr{O}_{K_0}^{+}/\mathrm{N}_{K/K_0}(\mathscr{O}_K^{*})$. Let ε_0 and ε be the fundamental units of K_0 and K, respectively. If $\mathrm{N}(\varepsilon_0) = -1$, then $\mathscr{O}_{K_0}^{+} = \langle \varepsilon_0^2 \rangle = \mathrm{N}_{K/K_0}(\langle \varepsilon_0 \rangle) \subseteq \mathrm{N}_{K/K_0}(\mathscr{O}_K^{*})$, and the quotient group is trivial. If $\mathrm{N}(\varepsilon_0) = +1$, then $\mathscr{O}_{K_0}^{+} = \langle \varepsilon_0 \rangle$, and since $\varepsilon_0^2 = \mathrm{N}_{K/K_0}(\varepsilon_0) \in \mathrm{N}_{K/K_0}(\mathscr{O}_K^{*})$, the quotient group is either trivial or $\langle \varepsilon_0 \rangle / \langle \varepsilon_0^2 \rangle$, in which case `bnfisunit` of PARI/GP can be used to compute the exponent of the unit.

Multiplication is straightforward and can be made more efficient by a *reduction* step that outputs a smaller (not necessarily unique) representative. To reduce $(\mathfrak{a},\alpha)$, one computes an LLL-reduced ideal $\mathfrak{a}' = \mu\mathfrak{a}$ (using `idealred` in PARI/GP) and lets $\alpha' = \mu\overline{\mu}\alpha$ One then tries to reduce the unit contribution in the size of the algebraic number α' by multiplying it with an appropriate power of $\mathrm{N}_{K/K_0}(\varepsilon)$.

The Shimura group $\mathfrak{C}$ and its subgroup $\mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K_r})$ can be enumerated directly; but the map $\mathrm{N}_{\Phi^r} : \mathrm{Cl}_{K_r} \to \mathfrak{C}$ being in general non-injective, this can require a large number of expensive principality tests in $\mathfrak{C}$ to avoid duplicates. More elegantly, we may consider the groups in (2.14) as given by cyclic bases or, more generally, generators and relations, and complete the sequence from known data using tools of linear algebra for $\mathbb{Z}$-modules, in particular the Hermite (HNF) and Smith normal forms (SNF), see [Coh93, §2.4].

Algorithm 2.3.1 :

INPUT: Cyclic bases for Cl_K and $\mathrm{Cl}^+_{K_0}$

OUTPUT: Cyclic basis for $\mathfrak{C}$

1) Compute a matrix M for $\mathrm{N}_{K/K_0} : \mathrm{Cl}_K \to \mathrm{Cl}^+_{K_0}$.
2) Compute generators $\mathfrak{a}_1, \ldots, \mathfrak{a}_r$ of the kernel of M.
3) Lift $\mathfrak{a}_1, \ldots, \mathfrak{a}_r$ to $\mathfrak{C}$: Pick arbitrary totally positive $\alpha_i \in K_0$ such that $\mathfrak{a}_i \overline{\mathfrak{a}}_i = \alpha_i \mathcal{O}_{K_0}$.
4) Compute a basis for the lattice L_0 of relations such that the subgroup of Cl_K generated by $\mathfrak{a}_1, \ldots, \mathfrak{a}_r$ is isomorphic to $\mathbb{Z}^r / L_0$.
5) If $\mathcal{O}^+_{K_0} / \mathrm{N}_{K/K_0}(\mathcal{O}^*) = 1$, let $r' = r$; otherwise let $r' = r + 1$ and $(\mathfrak{a}_{r'}, \alpha_{r'}) = (\mathcal{O}_K, \varepsilon_0)$.
6) Expand the basis of 4) into a basis for the lattice L of relations between $(\mathfrak{a}_1, \alpha_1), \ldots, (\mathfrak{a}_{r'}, \alpha_{r'})$ such that $\mathfrak{C} \simeq \mathbb{Z}^{r'} / L$.
7) Determine a cyclic basis of $\mathfrak{C}$. ◇

Step 1) requires to apply the generalised discrete logarithm map in $\mathrm{Cl}^+_{K_0}$ to the small number of relative norms of the basis elements of Cl_K. Step 3) is possible since the $\mathfrak{a}_i \overline{\mathfrak{a}}_i = \mathrm{N}_{K/K_0}(\mathfrak{a}_i)$ are trivial in $\mathrm{Cl}^+_{K_0}$. Steps 5) and 6) rely on the exactness of the sequence (2.14). If $r' = r$, there is nothing to do. Otherwise, we first add the relation $(\mathcal{O}_K, \varepsilon_0)^2 = 1$. Lifts of relations from L_0 are then in the image of $\langle \varepsilon_0 \rangle / \langle \varepsilon_0^2 \rangle$, and if the unit exponent is odd in the lift, we need to add $(\mathcal{O}_K, \varepsilon_0)$ into the relation. Steps 2) and 4) require an HNF, Step 7) an SNF.

2.3.2 The type norm subgroup

Algorithm 2.3.1 also provides an algorithm for generalised discrete logarithms in $\mathfrak{C}$, which can be used to determine the subgroup $N_{\Phi^r}(\mathrm{Cl}_{K^r})$ in a similar way: For each generator of Cl_{K^r}, we compute the generalised discrete logarithm of its image in $\mathfrak{C}$, then the relations between the images using an HNF and a cyclic basis using an SNF. The enumeration of the subgroup is then trivial. In the same vein, it is possible to compute all the cosets $\mathfrak{C} / N_{\Phi^r}(\mathrm{Cl}_{K^r})$ if the complete Igusa class polynomial is desired and not only its irreducible factor H_1, see §2.1.3.

3 Reducing the class polynomials

3.1 CM theory and reduction

The analogue of Theorem 1.2.2 is given by Shimura:

Theorem 3.1.1 : *Let p be a non ramified prime in the Galois closure of K, and let $A/\mathfrak{H}$ be an abelian variety with CM by $\mathcal{O}_K$. Then A has (potentially) good reduction modulo p.*

More precisely, let $\mathfrak{p}$ be a prime of degree 1 above p in $\mathcal{O}_{K^r}$, and let $\mathfrak{P}$ be a prime above $\mathfrak{p}$ in $\mathfrak{H}$. Then the Artin symbol $\left(\frac{\mathfrak{P}}{\mathfrak{p}}\right)$ corresponds to the action of the Frobenius of $\mathbb{F}_{\mathfrak{p}}$ on $A_{\mathfrak{P}}$ and is given by the action of the type norm $\mathfrak{a}$ of $N_{\mathfrak{H}/K^r}(\mathfrak{P})$ in the Shimura class group. In particular, A reduces to $\mathbb{F}_{\mathfrak{p}^n}$ where n is the order of the type norm in the Shimura class group.

Furthermore, we can write $\mathfrak{a}^n = (\pi)$ where $\pi\overline{\pi} = p^n$ (this condition determines π up to roots of unity). Then $\pi \in \mathcal{O}_K$ corresponds to the action of the Frobenius of $\mathbb{F}_{\mathfrak{p}^n}$ on $A_{\mathfrak{P}}$, and its characteristic polynomial gives the Weil polynomial of the corresponding isogeny class.

Proof: See [Shi98, §III.13], [Str10, Theorem 4.1] or [Mil06, Theorem 8.1]. ■

For $\mathfrak{p} \in \mathcal{O}_{K^r}$, by abuse of notation we will often speak of reduction modulo $\mathfrak{p}$ for the reduction of an abelian variety $A/\mathfrak{H}$ modulo a prime $\mathfrak{P} \in \mathfrak{H}$ above $\mathfrak{p}$. Since $\mathfrak{H}/K_0^r$ is Galois, we will also speak of reduction modulo $\mathfrak{p}_0 = \mathfrak{p} \cap K_0^r$. Finally, when $\mathfrak{p}_0$ if of degree 1 above p, we have $\mathbb{F}_{\mathfrak{p}_0} = \mathbb{F}_p$, so we will speak of reduction modulo p.

The type of the reduction of the CM abelian variety A according to the decomposition of p in the Galois closure of K is decribed in [GL12]. We give an overview of the type with good reduction, which corresponds to the case without ramification. We note $\mathfrak{p}$ an element of $\mathcal{O}_{K_0}$ above p. We recall that a supersingular abelian variety over a finite field of characteristic p is isogenous to a product of supersingular elliptic curves (equivalently, the p-rank of A is of dimension 0), and a superspecial variety is isomorphic to a product of supersingular elliptic curves. Finally, A is said to be ordinary if the p-rank is equal to the dimension of A.

Theorem 3.1.2 : *Let K be cyclic. Then we have the following cases:*

1) $\mathfrak{p}$ is of degree 1 over p and splits in $\mathcal{O}_K$. Then A has ordinary reduction modulo $\mathfrak{p}$.

2) $\mathfrak{p}$ is of degree 1 over p and is inert in $\mathcal{O}_K$. Then A is superspecial modulo $\mathfrak{p}$.

3) $\mathfrak{p}$ is inert over p. Then it is also inert in $\mathcal{O}_K$, and the reduction of A modulo $\mathfrak{p}$ is supersingular, but not superspecial.

Let K be dihedral. We have the following cases:

1) $\mathfrak{p}$ is of degree 1 over p and splits in $\mathcal{O}_{K^r}$. Then A has ordinary reduction modulo $\mathfrak{p}$.

2) $\mathfrak{p}$ is of degree 1 over p and is inert in $\mathcal{O}_{K^r}$. Then A is superspecial modulo $\mathfrak{p}$.

3) $\mathfrak{p}$ is inert over p and splits in $\mathcal{O}_{K^r}$. Then the reduction of A modulo $\mathfrak{p}$ has p-rank 1.

4) $\mathfrak{p}$ *is inert over* p *and is also inert in* $\mathcal{O}_{K^r}$*. Then the reduction of* A *modulo* $\mathfrak{p}$ *is supersingular, but not superspecial.*

Proof : [GL12, Table 3.3.1, Table 3.5.1]. ■

Remark 3.1.3 : The ramified case is as follow: if $\mathfrak{p}$ is ramified in $\mathcal{O}_{K^r}$, then A has superspecial reduction modulo $\mathfrak{p}$. If $\mathfrak{p}$ is not ramified in $\mathcal{O}_{K^r}$, but $\mathfrak{p}$ is ramified over p (this can only happen in the Dihedral case), then the reduction of A modulo $\mathfrak{p}$ has p-rank 1. ◇

3.2 CRT primes

A *CRT prime* $\mathfrak{p} \subset \mathcal{O}_{K_0^r}$ is a prime such that all abelian surfaces over $\mathbb{C}$ with CM by $(\mathcal{O}_K, \Phi)$ have ordinary good reduction of degree 1 modulo $\mathfrak{p}$. By Theorems 3.1.1 and 3.1.2, $\mathfrak{p}$ is a CRT prime for the CM-type Φ if and only if there exists an unramified prime $\mathfrak{q}$ in $\mathcal{O}_{K^r}$ of degree 1 above $p = \mathfrak{p} \cap \mathbb{Z}$ of principal type norm (π) with $\pi\overline{\pi} = N_{K/\mathbb{Q}}(\mathfrak{q})$; in particular, this implies that $\mathfrak{q}$ is totally split in the class field corresponding to the abelian surfaces with CM by $(\mathcal{O}_K, \Phi)$.

By abuse of notation, since $\mathfrak{p}$ has to be of degree 1 over p, we also say that p is a CRT prime for the CM type Φ.

Theorem 3.2.1 : *Let* Φ *be a CM type and* $p \in \mathbb{Z}$ *a prime number. If there exists an unramified prime* $\mathfrak{p}$ *in* $\mathcal{O}_{K^r}$ *of degree* 1 *above* p *of principal type norm* (π) *such that* $\pi\overline{\pi} = p$*, then* p *is a CRT prime for the CM type* Φ*.*

In this case, the reduction of the class polynomials defined over $\mathcal{O}_{K_0^r}$ *are completely split over* $\mathbb{F}_p$ *with roots of multiplicity* 1*.*

If moreover p *splits completely in* $\mathcal{O}_{K^r}$ *into primes with principal type norm, then* p *is a CRT prime for both CM types. In this case, the full class polynomials defined over* $\mathbb{Q}$ *are completely split over* $\mathbb{F}_p$ *with roots of multiplicity one.*

Proof : Let $A/\mathbb{C}$ be an abelian variety of CM type (K, Φ). There exists a model of A defined over the field $\mathfrak{H}$ generated by the Igusa invariants of A (see Theorem 1.3.1). By Theorem 3.1.1, the condition on $\mathfrak{p}$ implies that it splits completely in this field (more precisely, the type norm of $\mathfrak{p}$ gives the action of the Frobenius in terms of the Shimura class group; the abelian variety $A_{\mathfrak{p}}$ descends to $\mathbb{F}_p$ if and only if this element is trivial). If we fix one of the primes $\mathfrak{P}$ above $\mathfrak{p}$, then by Theorem 3.1.2 $A_{\mathfrak{P}}$ has ordinary reduction defined over $\mathbb{F}_p$. All such abelian varieties with CM by Φ have ordinary reduction over $\mathbb{F}_p$, so p is a CRT prime. Now all the roots have multiplicity 1 because the abelian varieties defined over $\mathfrak{H}$ are the canonical lifts of their ordinary reduction. Since they are not isomorphic over $\mathfrak{H}$, their reduction are not isomorphic over $\mathbb{F}_p$.

For the second assertion, we have by the hypothesis that p splits completely as $p = \mathfrak{p}_1\mathfrak{p}_2$ in $\mathcal{O}_{K_0^r}$ (which does not depend of the CM type). By the preceding paragraph, since all primes in $\mathcal{O}_{K^r}$ above $\mathfrak{p}_1$ satisfy the condition of the first part, p is a CRT prime for the CM type Φ. This concludes the cyclic case. Now for the dihedral case, let Φ' be the other CM type. We will see in §3.3 that the reductions of the abelian varieties with CM by Φ modulo primes in $\mathcal{O}_{K^r}$ above $\mathfrak{p}_2$ correspond exactly to the reductions of the abelian varieties with CM by Φ' modulo primes in $\mathcal{O}_{K_{\Phi'}}$ above $\mathfrak{p}_1$. So p is also a CRT prime for Φ'. ■

Remark 3.2.2 :

- When we speak of roots of multiplicity 1, we mean roots of the system of dimension 0 defined by H_1, $\widehat{H}_2$ and $\widehat{H}_3$. The polynomial $H_1 \mod p$ may have multiple roots (they correspond to abelian varieties with the same j_1-invariant modulo p but different j_2, j_3).
- If p is a prime such that the reductions are supersingular, the roots can appear with multiplicity. An example can be found in [GL12, Section 3.6.6]. ◊

3.2.1 The cyclic case

In the cyclic case, there is just one CM type Φ, 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 type norm. (Since the other primes are all conjugate to this one, they all also have principal type norms, and give the same isogeny class.)

3.2.2 The dihedral case

For the dihedral case, a prime number p is a CRT prime if it splits completely as $\mathfrak{q}_1\mathfrak{q}_2$ in K_0^r, such that $\mathfrak{q}_1$ splits completely in K^r into primes with principal type norm (π) and $\pi\overline{\pi} = p$.

For the CRT algorithm described in Chapter 7, when we compute the resulting interpolating Igusa polynomials over $\mathbb{F}_p$, they are the reduction of the Igusa class polynomials modulo $\mathfrak{q}_1$ and the CRT step will then be computed over $\mathcal{O}_{K_0^r}$, not over $\mathbb{Z}$ as in the cyclic case. We also note that if $\mathfrak{q}_2$ also splits completely into prime ideals with principal type norm, we could use $\mathfrak{q}_2$ in the CRT rather than $\mathfrak{q}_1$. We can then use the sieving algorithms of §7.2 to select the one most practical for the computation.

3.3 The other CM type

In the Dihedral case we have fixed once and for all a CM type and the corresponding reflex field K^r. One would think that to compute class polynomials for the other CM type, one would need to change K^r accordingly. In fact, since the two CM types are conjugate under the action of $\mathrm{Gal}(K_0^r/\mathbb{Q})$ we can work directly over K^r to handle both CM types. This section explains how.

Let Φ_1 and Φ_2 be the two distinct CM types Φ_1 and Φ_2. Let A_1 (resp. A_2) be an abelian variety with complex multiplication by Φ_1 (resp. Φ_2) and p a prime that splits as $\mathfrak{q}_1\mathfrak{q}_2$ in K_0^r (this field does not depend on the choice of the CM type).

Note that since the two CM types are conjugate, this gives that $(A_2)_{\mathfrak{q}_1}$, where by abuse of notation we denote by $(A_2)_{\mathfrak{q}_1}$ the reduction of A_2 modulo a prime above $\mathfrak{q}_1$ in the moduli field where A_2 is defined, is isogenous to $(A_1)_{\mathfrak{q}_2}$. Thus both CM types can be handled at once by fixing once and for all the reflex field (corresponding to one fixed CM type Φ) and looking at both primes $\mathfrak{q}_1$ and $\mathfrak{q}_2$. In other words, the reduction modulo $\mathfrak{q}_1$ for the second CM type can be obtained by looking at the reduction of the curves with CM by the first CM type modulo $\mathfrak{q}_2$.

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. For instance, to compute the class polynomial over $\mathbb{Q}$ using the CRT algorithm (i.e. corresponding to all curves with CM by $\mathcal{O}_K$), we have to look for CRT primes p that split completely in K_0^r as $p = \mathfrak{q}_1\mathfrak{q}_2$ and such that both $\mathfrak{q}_1$ and $\mathfrak{q}_2$ split completely, and the primes above them have principal type norm. (The corresponding type norm of the primes above $\mathfrak{q}_1$ and $\mathfrak{q}_2$ will then give the action of Frobenius on both isogeny classes).

3.4 The Weil polynomial corresponding to the isogeny class

Once we have computed the class polynomials, we want to compute the number of points of an abelian variety with CM by $\mathcal{O}_K$ reduced modulo a certain prime number p. This number is easily derived from the Weil polynomial of the Frobenius π from Theorem 3.1.1.

Likewise, the CRT algorithm in Chapter 7 needs to find, for a given CRT prime p, 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 one isogeny class (up to twists). 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 characterisation of this isogeny class. By a theorem of Tate, the isogeny class is characterised by the zeta function of a curve in it, or equivalently by the characteristic polynomial (the Weil polynomial) of the action of the Frobenius $\pi \in K$.

Similarly, the p-adic method of Chapter 6 will try to find a curve with CM by $\mathcal{O}_K$ over $\mathbb{F}_p$; once found, this curve will be lifted over $\mathbb{Q}_p$.

In all cases 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 Φ, then Theorem 3.1.1 shows that the type norm of $\mathfrak{p}$ gives the action of the Frobenius. This yields the following algorithm:

Algorithm 3.4.1 : Characterising the isogeny class.

```
INPUT: An unramified prime $\mathfrak{p}$ (above the prime $p \in \mathbb{Z}$) in $\mathcal{O}_{K^r}$ of degree 1 of principal type norm.
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 $\mathrm{N}_{\Phi^r}(\mathfrak{p}) = (\alpha)$.
    2) Compute the fundamental unit $\xi$ of $K_0$.
    3) Choose an embedding $K \mapsto \mathbb{C}$ and let $|\cdot|$ be the corresponding absolute value.
    4) Compute $n = \log(p/|\alpha|^2)/\log(|\xi|^2) \in \mathbb{Z}$.
    5) Compute $\pi = \alpha \xi^n$.
    6) Return the characteristic polynomial of $\pi$.                                ◇
```

Remark 3.4.2 : The ideal generated by α has relative norm p (by definition of the type norm), but we need to find a generator π with complex absolute value equal to $\sqrt{p}$. We adjust α by multiplying by some power of the fundamental unit ξ in K_0.

Note that π is only well-defined up to a root of unity. These roots of unity correspond to twists of the abelian varieties. Note that apart from the Galois field $\mathbb{Q}(\zeta_5)$ containing the 5-th root of unity, for the other CM fields the only roots of unity are ±1, corresponding to quadratic twists. ◇

Let χ_π be the Weil polynomial. We recall that χ_π is of the form $X^4 - tX^3 + sX^2 - tqX + q^2$. If $\mathrm{Jac}(C)$ has χ_π as the reciprocal of the numerator of its zeta function, then $\#C(\mathbb{F}_q) = 1 + q - t$ and $\#J(\mathbb{F}_q) = \chi_\pi(1) = 1 - t + s - tq + q^2$. Conversely, if $m = \#C(\mathbb{F}_q)$ and $n = \#J(\mathbb{F}_q)$ (for instance $n = (m^2 + m_2)/2 - q$ where $m_2 = \#C(\mathbb{F}_{q^2})$), then $t = 1 + q - m$ and $s = n - 1 + t + tq - q^2$.

4 The analytic method

4.1 Algorithm for Igusa class polynomials

We briefly summarise the algorithm for computing class polynomials.

Algorithm 4.1.1 :

INPUT: CM field K and CM type $\Phi = \{\varphi_1, \varphi_2\}$ of K

OUTPUT: Irreducible class polynomials H_1, $\widehat{H}_2$, $\widehat{H}_3 \in K_0^r[X]$ in the dihedral case and $\in \mathbb{Q}[X]$ in the Galois case

1) Compute $\mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) = \{(\mathfrak{b}_1, u_1), \ldots, (\mathfrak{b}_h, u_h)\} \subseteq \mathfrak{C}$.

2) Compute a base point $\beta = (\Phi, \mathfrak{a}_\Phi, \xi_\Phi)$ such that $\begin{cases} (\mathfrak{a}_\Phi \overline{\mathfrak{a}}_\Phi \delta_K)^{-1} = (\xi_\Phi), \\ \varphi_1(\xi_\Phi), \varphi_2(\xi_\Phi) \in i\mathbb{R}^{>0}. \end{cases}$

3) Enumerate $\{C \cdot \beta = (\Phi, \mathfrak{b}_i^{-1} \mathfrak{a}_\Phi, u_i \xi_\Phi),\ C = (\mathfrak{b}_i, u_i) \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})\}$ and compute the associated period matrices $\Omega_i = \Omega_{C \cdot \beta}$ for $i = 1, \ldots, h$.

4) For $i = 1, \ldots, h$, compute the fundamental ϑ-constants $\vartheta_0(\Omega_i/2), \ldots, \vartheta_3(\Omega_i/2)$; then deduce the squares of the ten even ϑ-constants $\vartheta_k^2(\Omega_i)$ by (2.6), the values $h_k(\Omega_i)$ by (2.7) and finally the triples $J_i = \big(j_1(\Omega_i), j_2(\Omega_i), j_3(\Omega_i)\big)$ by (2.8).

5) Let $H_1 = \prod_{i=1}^h (X - J_{i,1})$, $\widehat{H}_k = \sum_{i=1}^h J_{i,k} \prod_{l \neq i} (X - J_{l,1}) \in \mathbb{C}[X]$ for $k \in \{2, 3\}$.

6) Recognise the coefficients of H_1, $\widehat{H}_2$, $\widehat{H}_3$ as elements of K_0^r or $\mathbb{Q}$, respectively. ♢

The different steps of the algorithm and our implementation are detailed in the following chapters. The symbolic computations related to number fields in Steps 1) and 2) and to the period matrices Ω_i in Step 3) were described in §2.2. Step 1) was treated in §2.3.2, Step 4) is treated in §4.2 and Step 6) in §4.3.

4.2 Computing ϑ-constants

As explained in Step 4) of Algorithm 4.1.1, it suffices to compute the fundamental ϑ-constants $\vartheta_0, \ldots, \vartheta_3$ in the argument $\Omega/2$ to obtain the class invariants for the period matrix $\Omega = \begin{pmatrix} \omega_0 & \omega_1 \\ \omega_2 & \omega_0 \end{pmatrix} \in \mathscr{F}_2$.

In §4.2.1 we describe an algorithm to compute the ϑ-constants directly from their q-expansions, using a lower number of multiplications than approaches described previously in the literature.

As the coefficients of the Igusa class polynomials grow rather quickly, a high floating point precision is needed for evaluating the ϑ-constants. In §§4.2.2–4.2.4 we describe an algorithm with a quasi-linear (up to logarithmic factors) complexity in the desired precision, using Newton iterations on a function involving the Borchardt mean. The algorithm is described essentially in Dupont's PhD thesis [Dup06]; for the corresponding algorithm in dimension 1, using the arithmetic-geometric mean instead of the Borchardt mean, see [Dup11]. We provide a streamlined presentation in dimension 2, together with improved algorithms and justifications.

4.2.1 Naive approach

For the fundamental ϑ-constants, (2.5) specialises as

$$\vartheta_{4b_1+2b_2}(\Omega/2) = \sum_{m,n\in\mathbb{Z}} (-1)^{2(mb_1+nb_2)} q_0^{m^2} q_1^{2mn} q_2^{n^2} \tag{4.1}$$

with $q_k = \exp(i\pi\omega_k/2)$.

Positive definiteness and reducedness of the binary quadratic form attached to $\Im(\Omega)$ show that the sum converges when taken over, for instance, a square $[-R,R]^2$ with $R\to\infty$; [Dup06, p. 210 following the proof of Lemma 10.1, with typos] establishes that for $R \geqslant \sqrt{1.02N+5.43}$, the truncated sum is accurate to N bits. Better bounds may be reached using summation areas related to the eigenvalues of $\Im(\Omega)$, but using a square allows to organise and reuse computations so as to reduce the number of complex multiplications.

Proposition 4.2.1 : *The truncated sum over $(m,n)\in[-R,R]^2$ for the fundamental ϑ-constants (4.1) may be computed with $2R^2+O(R)$ multiplications and one inversion using storage for $R+O(1)$ elements.*

Letting $R = \lceil\sqrt{1.02N+5.43}\rceil$ and using complex numbers of precision $O(N)$, we obtain a time complexity of

$$O(N\,\mathrm{M}(N)) \text{ or } \widetilde{O}(N^2),$$

where $\widetilde{O}(N) = O\left(N(\log N)^{O(1)}\right)$, and $\mathrm{M}(N)\in\widetilde{O}(N)$ is the time complexity of multiplying two numbers of N bits.

Proof : Using symmetries with respect to the signs of m and n, we may write

$$\begin{aligned}\sum_{-R\leqslant m,n\leqslant R} (-1)^{2(mb_1+nb_2)} q_0^{m^2} q_1^{2mn} q_2^{n^2} &= 1+2\sum_{m=1}^{R}(-1)^{2mb_1}q_0^{m^2}+2\sum_{n=1}^{R}(-1)^{2nb_2}q_2^{n^2}\\ &\quad +2\sum_{m=1}^{R}(-1)^{2mb_1}q_0^{m_2}\sum_{n=1}^{R}(-1)^{2nb_2}q_2^{n^2}\left(q_1^{2mn}+q_1^{-2mn}\right).\end{aligned}$$

We first compute and store the $q_2^{n^2}$ with $2R+O(1)$ multiplications via $q_2^{2n-1} = q_2^{2(n-1)-1}\cdot q_2^2$ and $q_2^{n^2} = q_2^{(n-1)^2}\cdot q_2^{2n-1}$. After computing the inverse q_1^{-1}, a similar scheme yields the $q_0^{m^2}$ and $q_1^{2m}+q_1^{-2m}$ without storing them. At the same time, we may compute for any given m the sum over n inside the double sum: The term $q_1^{2mn}+q_1^{-2mn}$ is the n-th element v_n of the Lucas sequence $v_0=2$, $v_1 = q_1^{2m}+q_1^{-2m}$, $v_n = v_1\cdot v_{n-1}-v_{n-2}$, each element of which is computed with one multiplication. Together with the multiplication by $q_2^{n^2}$, each term of the innermost sum is thus obtained with two multiplications.

For the time complexity, recall that complex inversions can be computed in time $O(\mathrm{M}(N))$, and exponentials in time $O(\mathrm{M}(N)\log N)$, see [Bre76]. ■

This algorithm gains an asymptotic factor of 2/3 over [Dup06, Algorithme 15].

4.2.2 Borchardt mean of complex numbers

The key tool in the asymptotically fast evaluation of ϑ-constants is the Borchardt mean, which generalises Lagrange's and Gauß's arithmetic-geometric mean of two numbers to four. The Borchardt mean of four positive real numbers has been introduced in [Bor76; Bor78]. The complex case is treated in [Dup06], where proofs of most (but not all) propositions below may be found. It is made complicated by the presence of several square roots in the formulæ, each of which is defined only up to sign.

Definition 4.2.2 : Let

$$\begin{aligned}\mathscr{H} &= \left\{z \in \mathbb{C} : \arg(z) \in \left]-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} \cup \{0\} \\ &= \{z \in \mathbb{C} : \Re(z) > 0, \text{ or } \Re(z) = 0 \text{ and } \Im(z) \geqslant 0\}\end{aligned}$$

be the *complex half-plane* defining the standard branch of the complex square root function. For a number in $\mathscr{H}$, its square root in $\mathscr{H}$ lies in fact in the *complex quarter-plane*

$$\mathscr{Q} = \left\{z \in \mathbb{C} : \arg(z) \in \left]-\frac{\pi}{4}, \frac{\pi}{4}\right]\right\} \cup \{0\}.$$

Definition and Properties 4.2.3 : Given a complex quadruple $b = (b_0, \ldots, b_3) \in \mathbb{C}^4$, a *Borchardt iterate* is a quadruple $b' = (b'_0, \ldots, b'_4)$ such that there are four choices of square roots $(\sqrt{b_j})_{j=0,\ldots,3}$ yielding

$$b'_0 = \tfrac{1}{4}(b_0 + b_1 + b_2 + b_3) \qquad b'_1 = \tfrac{1}{2}(\sqrt{b_0}\sqrt{b_1} + \sqrt{b_2}\sqrt{b_3})$$
$$b'_2 = \tfrac{1}{2}(\sqrt{b_0}\sqrt{b_2} + \sqrt{b_1}\sqrt{b_3}) \qquad b'_3 = \tfrac{1}{2}(\sqrt{b_0}\sqrt{b_3} + \sqrt{b_1}\sqrt{b_2})$$

There are up to eight different Borchardt iterates of a given quadruple. If $b \in \mathscr{H}^4$, the *standard Borchardt iterate* is obtained by choosing square roots in $\mathscr{Q}$, so that $b' \in \mathscr{H}^4$ again. More generally, if all entries of b lie in the same half-plane, that is, $b \in (z\mathscr{H})^4$ for some $z \in \mathbb{C}$, choosing all square roots in the same quarter-plane $\sqrt{z}\mathscr{Q}$ (with either choice of sign for $\sqrt{z}$) yields the standard Borchardt iterate in the same half-plane.

A *Borchardt sequence* is a sequence $\left(b^{(n)}\right)_{n \geqslant 0}$ such that $b^{(n+1)}$ is a Borchardt iterate of $b^{(n)}$ for all $n \geqslant 0$. If all entries of $b^{(0)}$ lie in the same half-plane, its *standard Borchardt sequence* is defined by taking only standard Borchardt iterates.

The following result is proved in [Dup06, Chapter 7].

Proposition 4.2.4 : *Any Borchardt sequence converges to a limit* (z, z, z, z).

When the elements of b *are contained in the same half-plane, the* Borchardt mean $B_2(b)$ *of* b *is the limit of the standard Borchardt sequence starting with* $b^{(0)} = b$*. The function* B_2 *is obviously homogeneous.*

A standard Borchardt sequence converges quadratically:

$$\left\| b^{(n)} - B_2(b) = 2^{-O(2^n)} \right\|.$$

This implies that the Borchardt mean is computed to a of precision N *bits with* $O(\log N)$ *multiplications in time*

$$O(\mathrm{M}(N) \log N).$$

Comparison of the formulæ in Definition 4.2.3 and (2.6) shows that for any period matrix $\Omega \in \mathscr{H}_2$, the sequence $\left((\vartheta_j^2(2^n\Omega))_{j=0,\ldots,3}\right)_{n\geqslant 0}$ is a Borchardt sequence. This fact alone does not solve the sign issue, however. One would hope for the ϑ-sequence to be the standard Borchardt sequence, which would allow it to be computed with the standard choice of complex square roots. This assumption does not hold in general; however, it is true for the fundamental ϑ-constants and $\Omega \in \mathscr{F}_2$.

Proposition 4.2.5 : *For $\Omega \in \mathscr{F}_2$, $n \geqslant 0$ and $j = 0,\ldots,3$ we have $\vartheta_j(2^n\Omega) \in \mathscr{Q}$. Hence $\left((\vartheta_j^2(2^n\Omega))_{j=0,\ldots,3}\right)_{n\geqslant 0}$ is the standard Borchardt sequence associated to $(\vartheta_j^2(\Omega))_{j=0,\ldots,3}$. It converges to 1.*

The result follows from [Dup06, Propositions 6.1 and 9.1].

4.2.3 Period matrix coefficients from ϑ-constants

For the time being, we consider the inverse of the function we are interested in and describe an algorithm that upon input of the values of the four fundamental ϑ-quotients in a period matrix returns the coefficients of the period matrix. Newton iterations can then be used to invert this function.

By the modularity of the squares of the ϑ-constants, applying a matrix $\gamma \in \mathrm{Sp}_4(\mathbb{Z})$ to their argument Ω permutes the functions and multiplies them by a common projective factor, which depends on γ and Ω. In this way, information on Ω can be gathered; informally, three matrices suffice to obtain the three different coefficients of Ω. We consider three particular matrices, as suggested in [Dup06, §9.2.3], which lead to well-behaved Borchardt means, see Conjecture 4.2.7.

Proposition 4.2.6 : *Let $\mathfrak{J} = \begin{pmatrix} 0 & -\mathrm{id}_2 \\ \mathrm{id}_2 & 0 \end{pmatrix}$ and $\mathfrak{M}_j = \begin{pmatrix} \mathrm{id}_2 & m_j \\ 0 & \mathrm{id}_2 \end{pmatrix}$ with $m_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $m_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $m_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. Let $\Omega \in \mathscr{H}_2$. Then*

$$\begin{aligned}
\left(\vartheta_j^2((\mathfrak{J}\mathfrak{M}_0)^2\Omega)\right)_{j=0,1,2,3} &= -i\omega_0\left(\vartheta_j^2(\Omega)\right)_{j=4,0,6,2},\\
\left(\vartheta_j^2((\mathfrak{J}\mathfrak{M}_1)^2\Omega)\right)_{j=0,1,2,3} &= (\omega_1^2-\omega_0\omega_2)\left(\vartheta_j^2(\Omega)\right)_{j=0,8,4,12},\\
\left(\vartheta_j^2((\mathfrak{J}\mathfrak{M}_2)^2\Omega)\right)_{j=0,1,2,3} &= -i\omega_2\left(\vartheta_j^2(\Omega)\right)_{j=8,9,0,1}.
\end{aligned}$$

A more general statement with the action on the ϑ-constants (not squared) is given in [Cos11, Propriété 3.1.24], following [Igu72, Chapter 5, Theorem 2]. The explicit form restricted to squares of ϑ-constants, as given here, is found in [Dup06, §6.3.1].

The idea of the algorithm is now to apply the Borchardt mean function B_2 to both sides of the above equations. Conjecturally, the left hand side becomes 1, so that each Borchardt mean of a right hand side yields a coefficient of Ω. So we rely on the following conjecture, for which we have overwhelming numerical evidence, but no complete proof. Notice that it is *a priori* not even clear if the Borchardt means are well-defined, that is, if the squares of the various four ϑ-values always lie in the same half-plane.

Conjecture 4.2.7 : *Let*

$$\mathscr{U} = \left\{\Omega \in \mathscr{H}_2 : B_2\left((\vartheta_j^2(\Omega))_{j=0,\ldots,3}\right) \text{ is defined and equal to } 1\right\}.$$

For $k \in \{0,1,2\}$ we have $(\mathfrak{J}\mathfrak{M}_k)^2\mathscr{F}_2 \subseteq \mathscr{U}$.

Under Conjecture 4.2.7, we can now formulate an algorithm to obtain Ω from four values of ϑ-constants. To make the following Newton iterations more efficient, we dehomogenise all modular functions by dividing by appropriate powers of ϑ_0, which allows to work with only three inputs.

Algorithm 4.2.8 :

INPUT: Floating point approximations of $\left(\vartheta_j(\Omega/2)/\vartheta_0(\Omega/2)\right)_{j=1,2,3}$ for some $\Omega \in \mathscr{F}_2$, and as auxiliary data the sign of ω_1.

OUTPUT: Floating point approximations of the coefficients ω_0, ω_1, ω_2 of $\Omega \in \mathscr{F}_2$

1) Use the duplication formulæ (2.6) to compute $(\vartheta_j^2(\Omega)/\vartheta_0^2(\Omega/2))_{j=0,1,2,3,4,6,8,9,12,15}$.
2) Compute $B_2((\vartheta_j^2(\Omega)/\vartheta_0^2(\Omega/2))_{j=0,1,2,3}) = \frac{1}{\vartheta_0^2(\Omega/2)}$.
3) Deduce $(\vartheta_j^2(\Omega))_{j=0,1,2,3,4,6,8,9,12,15}$.
4) Compute

$$u_0 = B_2((\vartheta_j^2(\Omega))_{j=4,0,6,2}), \quad u_2 = B_2((\vartheta_j^2(\Omega))_{j=8,9,0,1}), \quad u_1 = B_2((\vartheta_j^2(\Omega))_{j=0,8,4,12}).$$

5) Return $\omega_0' = \frac{i}{u_0}$, $\omega_2' = \frac{i}{u_2}$ and $\omega_1' = \pm\sqrt{\frac{1}{u_1} + \omega_0'\omega_2'}$ with the appropriate sign. ◇

The correctness of Algorithm 4.2.8 under Conjecture 4.2.7 follows from the discussions above. Step 1) uses the homogeneity of (2.6), Step 2) the homogeneity of the Borchardt mean and Proposition 4.2.5. The validity of Step 5) follows from Proposition 4.2.6 under Conjecture 4.2.7, using again the homogeneity of the Borchardt mean.

Notice that Ω is only well-defined up to the subgroup of $\mathrm{Sp}_4(\mathbb{Z})$ for which the ϑ-constants are modular. Assuming $\Omega \in \mathscr{F}_2$, the fundamental domain for all of $\mathrm{Sp}_4(\mathbb{Z})$, it is necessarily unique; Conjecture 4.2.7 implies that this particular representative for Ω is indeed returned by the algorithm.

4.2.4 Newton lift for fundamental ϑ-constants

Denote by

$$F : \mathbb{C}^3 \to \mathbb{C}^3, \quad \left(\vartheta_j(\Omega/2)/\vartheta_0(\Omega/2)\right)_{j=1,2,3} \mapsto \Omega,$$

the function computed by Algorithm 4.2.8, and by

$$f : \mathscr{F}_2 \to \mathbb{C}^3, \quad \Omega \mapsto \left(\vartheta_j(\Omega/2)/\vartheta_0(\Omega/2)\right)_{j=1,2,3},$$

its inverse on $\mathscr{F}_2$ (where Ω is interpreted as the three-element vector $(\omega_0, \omega_1, \omega_2)$ and not as a four-element matrix).

We use Newton iterations on F to compute f. The standard Newton approach requires to compute the Jacobian matrix J_F of F, that is, its partial derivatives with respect to its different coordinates. Heuristically, Algorithm 4.2.8 may be modified accordingly to also output J_F, see [Dup06, Algorithme 16], generalising the dimension 1 approach of [BB87, §2.4] and [Dup11]. The description and justification of this algorithm are rather technical. Instead, we opt for using finite differences, which moreover turn out to yield a more efficient algorithm (see §8.1.1).

Algorithm 4.2.9 :

INPUT: Floating point approximations $y^{(n)}$ of $\Omega \in \mathscr{F}_2$, to precision $2N$ bits, and $x^{(n)}$ of $f(\Omega)$, to precision N bits.

OUTPUT: Floating point approximation $x^{(n+1)}$ of $f(\Omega)$, to precision $2N$ bits.

1) Let $\varepsilon = 2^{-N} \max_j \left\{ \left| x_j^{(n)} \right| \right\}$.
2) Let $(e_j)_{j=1,2,3}$ be the standard basis of $\mathbb{C}^3$. By Algorithm 4.2.8, compute $F(x^{(n)})$ and $\frac{\Delta F}{\Delta x_j} = \frac{1}{\varepsilon} \left(F(x^{(n)} + \varepsilon e_j) - F(x^{(n)}) \right)$.
3) Let $J = (J_{ij})_{i,j=1,2,3}$ with $J_{ij} = \frac{\Delta F_i}{\Delta x_j}$.
4) Let
$$x^{(n+1)} = x^{(n)} - \left(F(x^{(n)}) - y^{(n)} \right) J^{-1}$$
(where all vectors are seen as row vectors). ◇

All computations in the algorithm are carried out at a precision of $2N$ bits. But even without taking rounding errors into account, the approximation of the Jacobian matrix by finite differences as well as the Newton method itself introduce some inaccuracy in the result, so that the accuracy improves to less than $2N$ bits. The following proposition addresses this issue.

Theorem 4.2.10 : *Assume the validity of Conjecture 4.2.7. Let $\Omega \in \mathscr{F}_2$ be such that $\vartheta_0(\Omega/2) \neq 0$, $x = f(\Omega)$ and $x^{(0)}$ an initial floating point approximation to x. Not taking rounding errors into account, there exist two real numbers $\varepsilon_0 > 0$ and $\delta > 0$, depending on x, such that for $\|x^{(0)} - x\| < \varepsilon_0$, the sequence $x^{(n)}$ defined by successive applications of Algorithm 4.2.9 converges to x, with accuracy increasing in each step from N to $2N - \delta$.*

To reach a given accuracy N, the total complexity is dominated by the complexity of the last lifting step, that is:
$$O(\mathrm{M}(N) \log N).$$

Proof: By assumption, F is defined and analytic in a neighbourhood of x. In particular, its second partial derivatives are bounded in a close enough neighbourhood of x, so that the Jacobian matrix of F in $x^{(n)}$ is approximated to accuracy $2N - \delta_0$ bits by the matrix J computed in Steps 2) and 3), where δ_0 depends on x and on the bound on the second partial derivatives. The remaining assertion, with some $\delta \geqslant \delta_0$, is the standard result for Newton's method (see [GG99, Chapter 9 and §15.4] and [BZ10, §4.2]).

The complexity is derived from the superlinearity of multiplication, which makes the last of the $O(\log N)$ Newton steps dominate the whole computation; the logarithmic factor stems from the complexity of computing the Borchardt mean given in Proposition 4.2.4. ■

Notice that for the application of computing class polynomials for primitive quartic CM fields, the assumption of Theorem 4.2.10 is satisfied: As $\left(\Omega \quad \mathrm{id}_2\right) \mathbb{Z}^4$ is of rank 4, we have $\omega_1 \neq 0$, and therefore none of the $\vartheta_j(\Omega/2)$ vanish (see [Kli90, Chapter 9, Proposition 2]).

In practice, we use a fixed initial precision for $x^{(0)}$, computed according to Proposition 4.2.1, which determines ε and implicitly δ. The lack of information about δ can be worked around as follows: If $x^{(n-2)}$ and $x^{(n-1)}$ agree to k bits, and $x^{(n-1)}$ and $x^{(n)}$ agree to k' bits, we set $\delta = 2k - k'$. This value of δ accounts at the same time for bits lost due to rounding errors induced by the floating point computations.

Remark 4.2.11 : It is possible to modify Algorithm 4.2.8 and consequently Algorithm 4.2.9 so as not to rely on Conjecture 4.2.7. The conjecture states that the choices of square roots inside the Borchardt mean computations correspond to doubling the argument of the ϑ-constants. So by computing very low precision approximations of the ϑ-constants in $2^n\Omega$ as described in §4.2.1, one can make sure to choose the correct sign. These computations do not deteriorate the asymptotic complexity; moreover, as Algorithm 4.2.9 requires the Borchardt means of the same arguments over and over again (albeit with increasing precision), the sign choices may be fixed once and for all in a precomputation step.

In practice, however, we did not come upon any counterexample to Conjecture 4.2.7 with tens of thousands of arguments. ♢

4.3 Reconstruction of class polynomial coefficients and reduction modulo prime ideals

4.3.1 The dihedral case

The class polynomials H_1, $\widehat{H}_2$, $\widehat{H}_3$ of (2.9) and (2.10) for a fixed CM type Φ are defined over K_0^r, but Steps 1) to 5) of Algorithm 4.1.1 compute floating point approximations, precisely of the images of the polynomials under an embedding $\psi : K_0^r \to \mathbb{C}$. To realise Step 6) of Algorithm 4.1.1, we need to invert ψ: Given $\psi(\alpha)$ to sufficient precision, we wish to reconstruct α symbolically as an element of $K_0^r = \mathbb{Q}(z^r) = \mathbb{Q}[Z^r]/\left((Z^r)^2 + A^r Z^r + B^r\right)$, cf. §2.2.2. We may limit the discussion to the CM type Φ and the embedding ψ given by (2.13) and (2.12); the second CM type Φ' leads to class polynomials that are conjugate under $\mathrm{Gal}(K_0^r/\mathbb{Q})$.

Let D^r be the discriminant of K_0^r; as the discriminant of the minimal polynomial of z^r is $16B$, we have $K_0^r = \mathbb{Q}(\sqrt{B})$, and $\frac{D^r}{B}$ is a rational square. Let $w \in K_0^r$ with $w^2 = D^r$ satisfy $\psi(w) = \sqrt{D^r} > 0$. Write $\alpha = \frac{a+bw}{c}$ with coprime a, b, $c \in \mathbb{Z}$. Knowing an approximation β to $\psi(\alpha) \in \mathbb{R}$ at our working precision of n bits, we wish to recover a, b, c, for which there is hope if $2^n > |abc|$.

Let e be the exponent of β in the sense that $2^{e-1} \leqslant |\beta| < 2^e$, and let $e^+ = \max(e, 0)$ and $e^- = \max(-e, 0)$, so that $e = e^+ - e^-$, and at most one of e^+, e^- is non-zero. We expect $|a| \approx \sqrt{D^r}\,|b|$ (whereas c is usually smaller), so that $2^{e^-}|a| \approx 2^{e^-}\sqrt{D^r}\,|b| \approx 2^{e^-}|c\beta| \approx 2^{e^+}|c|$. On the other hand, the floating point approximation β satisfies $\left|\beta - \frac{a+b\sqrt{D^r}}{c}\right| \approx 2^{e-n}$ (up to a small factor accounting for digits lost to rounding errors), whence $2^{n+e^-}|c\beta - (a + b\sqrt{D^r}| \approx 2^{e^+}c$ is comparative in size to the previous quantities. Consider the integral matrix

$$\begin{pmatrix} 0 & 0 & 2^{n+e^+} \\ \left\lfloor 2^{e^-}\sqrt{D^r} \right\rceil & 0 & \left\lfloor 2^{n+e^+}\sqrt{D^r} \right\rceil \\ 0 & 2^{e^+} & \left\lfloor \beta 2^{n+e^+} \right\rceil \end{pmatrix}.$$

Using LLL, we find a short vector $\left(-b\left\lfloor 2^{e^-}\sqrt{D^r}\right\rceil, c2^{e^+}, r\right)$ in the lattice spanned by the rows of the matrix; the scaling of the last column was chosen, following the arguments above, such that all entries in the vector have comparable sizes. This determines b and c, and we let $a = \frac{c\left\lfloor \beta 2^{n+e^+}\right\rceil - b\left\lfloor 2^{n+e^+}\sqrt{D^r}\right\rceil}{2^{n+e^+}} \in \mathbb{Z}$.

To get back to our standard representation of K_0^r, we need to relate w and z^r. By (2.13) and (2.12),

$$\psi(z^r) = \psi(y^r)^2 = -A - 2\sqrt{B} < 0,$$

so that

$$w = \sqrt{\frac{D^r}{B}} \cdot \frac{-z^r - A}{2}. \tag{4.2}$$

To obtain abelian varieties over finite fields, we need to reduce the class polynomials modulo certain prime ideals $\mathfrak{p}_1$ of K_0^r. Let p be a rational prime that splits as $\mathfrak{p}_1\mathfrak{p}_2$ in K_0^r. Assume that $\mathfrak{p}_1$ splits in K^r, so that $\mathfrak{p}_1 = \mathfrak{q}_1\overline{\mathfrak{q}}_1$, and that the type norm of $\mathfrak{q}_1$ is a principal ideal of K. Then the class polynomial splits totally modulo $\mathfrak{p}_1$, and its reduction may be computed as follows: If $\mathfrak{p}_1 = p\mathcal{O}_{K_0^r} + (a + bw)\mathcal{O}_{K_0^r}$ with $a, b \in \mathbb{Z}$, replace each occurrence of w by $-\frac{a}{b}$ and reduce modulo p.

4.3.2 The cyclic case

Here the class polynomials are defined over $\mathbb{Q}$, its coefficients may be obtained by a 2-dimensional lattice reduction, and reduction modulo primes is trivial.

5 Isogenies and endomorphism rings

5.1 Introduction

If p is a CRT prime, the p-adic method to compute class polynomials needs to find one ordinary CM curve over $\mathbb{F}_p$ with CM by $\mathcal{O}_K$. It then lifts it over $\mathbb{Q}_p$ to a certain precision and computes the conjugate under the action of the type norm by explicit isogenies computations. Likewise, the CRT method needs to find one ordinary CM curve over $\mathbb{F}_p$ for several CRT primes p, and then computes the conjugates under the action of the whole Shimura class group (the reason one cannot compute an irreducible component modulo p is that it is hard to glue the right ones together across several CRT primes).

We see that they both need the same tools: First, find a curve in the right isogeny class. Then, once a curve is found, test if the endomorphism ring is maximal. Once a maximal curve is found, use “horizontal” isogenies (over $\mathbb{Q}_p$ or $\mathbb{F}_p$) to find the others. As in the genus 1 case, one can also use “vertical isogenies” to try to go from a curve in the right isogeny class into a maximal curve. This gives the following algorithm:

Algorithm 5.1.1 :

INPUT: A CRT prime p

OUTPUT: The Igusa invariants of an abelian variety over $\mathbb{F}_p$ with CM by $\mathcal{O}_K$.

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 this step fails, go back to Step 1. ◊

We will sometimes call the isogenies we compute in the going-up algorithm *vertical steps*, and the isogenies we compute in Algorithm 5.8.5 *horizontal steps*; this is in analogy with the corresponding terminology in the elliptic curve case.

We note that for both methods, rather than only look at primes of good ordinary reduction of degree 1, we could look at primes of good ordinary reduction of small degree d and work over $\mathbb{F}_q$ where $q = p^d$.

5.2 Searching for a curve in the isogeny class

First, once a CRT prime p is fixed, the associated isogeny class is obtained via Algorithm 3.4.1 where we determine π.

To find a curve in the isogeny class, one can either loop over the three Igusa invariants and reconstruct the curve from Mestre’s algorithm, or take random curves. While Mestre’s algorithm is quite slow, 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 $1/p^3$, so the probability of collision is quite low. This gives the following algorithm:

Algorithm 5.2.1 : Finding a curve in the isogeny class corresponding to $\pm\pi$.

If the characteristic polynomial of π 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 $\#\mathrm{Jac}(H)=N_1 := 1+t_0+t_1+t_2+p^2$. 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 $P\in\mathrm{Jac}(H)$. (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 5.2.2 : For the CRT method, note 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.

5.2.1 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 5.2.3 : *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) $\frac{\pi-1}{2}\in\mathcal{O}_K$.

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 π 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 λ, μ, ν does not matter, and there are 6 ways to permute those 3 roots.

Remark 5.2.4 : 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 that can be put 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 $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. $\diamondsuit$

5.2.2 Real multiplication

For a given quartic CM field K, write the real quadratic subfield $K_0 = \mathbb{Q}(\sqrt{d})$. For a given CRT prime p, any curve over $\mathbb{F}_p$ with CM by K must also have real multiplication by K_0. Such curves correspond to CM points associated to K on the Hilbert moduli space associated to K_0. They 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 [LY11]. Those formulas can be used directly in the search step for each CRT prime.

Here the same trade-off has to be considered as in the Rosenhain method: if $R \bigcap K_0 \not\supset \mathcal{O}_{K_0}$ 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.

5.3 Computing the ℓ-primary part of the torsion

Let A be an abelian surface over $\mathbb{F}_q$. To compute isogenies as in §5.4, we need to compute a basis of the points of the kernel. Likewise, to compute endomorphism rings via the method of §5.5, we need to compute the action of certain endomorphism on the ℓ^e-torsion. For both algorithm, we need to compute the ℓ-primary part of an abelian variety defined over an extension $\mathbb{F}_{q^d}$ of a certain degree d over $\mathbb{F}_q$.

We will use the following algorithm to compute points uniformly in the ℓ-primary group $A(\mathbb{F}_q)[\ell^\infty]$:

Algorithm 5.3.1 :

INPUT: An abelian surface $A/\mathbb{F}_q$ and a prime ℓ.

OUTPUT: Uniform random points in the group $A(\mathbb{F}_q)[\ell^\infty]$.

```
1) Precomputation:
    (a) Compute $\chi_{\pi^d}$ as the resultant in $X$ of $\chi_\pi(Y)$ and $Y^d - X$, and write $\#A(\mathbb{F}_{q^d}) = \chi_{\pi^d}(1) = \ell^e \gamma$ with $\gamma$ prime to $\ell$.
```

2) Repeat as needed:

   (a) Take a random point P (uniformly) in $A(\mathbb{F}_{q^d})$.

   (b) Return γP. ◇

Since the cardinal of $A(\mathbb{F}_{q^d})$ is roughly of q^{2d}, computing a random points take $O(d\log(q))$ operations in $\mathbb{F}_{q^d}$.

From uniform random points in $A(\mathbb{F}_{q^d})[\ell^\infty]$ one can look at the smallest k such that $\ell^k \cdot P$ is an ℓ^e-torsion point (with the notations of the algorithm); the problem here is that this method does not generate uniform random points of ℓ^e-torsion.

In a paper accompanying the source code of [BCR12] we explain how to do so by first computing an HNF basis of $A(\mathbb{F}_{q^d})[\ell^\infty]$, using Weil pairing to speed up the computation of thus a basis. From such a basis it is then trivial to recover a basis of the ℓ^e-torsion. First we give an example to illustrate it here.

Suppose that G is an ℓ-primary group generated by a point P of order ℓ^2 and a point Q of order ℓ. Assume that the first random point chosen is $P = R_1$, which gives an ℓ-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 ℓ-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 ℓ-torsion unless $\beta = 0$.

We give an overview of the algorithm

Algorithm 5.3.2 :

INPUT: A list G of generators of $A(\mathbb{F}_{q^d})[\ell^\infty]$.

OUTPUT: An HNF basis of $A(\mathbb{F}_q)[\ell^\infty]$.

```
1) Let $P_1,\dots,P_k$ be the current partial HNF basis, and let $n_i$ be the smallest number such
   that $Q_i = \ell^{n_i} P_i$ is a point of $\ell$-torsion. Repeat the following loop until we find the full
   HNF basis:
    (a) Take a new element $P \in G$.
    (b) Let $n$ be the smallest power such that $Q = \ell^n P$ is a point of $\ell$-torsion.
    (c) If Q is in the subgroup generated by the $Q_i$, write $Q = \sum \alpha_i Q_i$.
    (d) Switch $P_j$ and $P$ if $n_j$ is the minimal of $n$ and all the $n_i$ corresponding to points $Q_i$
        appearing in the support of the preceding decomposition.
    (e) Replace $P$ by $P - \sum \alpha_i \ell^{n_i - n} P_i$ and start the loop again.
    (f) If we are here then either $P = 0$ in which case we do not get an independent generator,
        or Q is not in the subgroup generated by the $Q_i$ so we add $P$ to the list. ◇
```

Remark 5.3.3 : In our setting we can use the Weil pairing to check if Q is in the group generated by the Q_i and compute the α_i in this case. This is done by constructing at each step of the preceding algorithm a matrix M such that M sends the Q_i to a “symplectic (partial) basis”. For instance, we known that we have the full basis of the ℓ^e-torsion if all n_i are greater or equal to $e-1$ and the matrix of the Weil pairing on the Q_i is non-degenerate.

The worse case is when we have an isotropic group (at worst of size $O(\ell^2)$), in which case we can use a baby step giant step approach to get an $O(\ell)$ complexity to find the "generalised logarithms". Since we have uniform points, we only need $O(1)$ random points to find a basis. The total cost is then $O(d \log p + \ell^2)$ operations if $\mathbb{F}_{q^d}$. $\diamondsuit$

5.4 On (ℓ,ℓ)-isogenies

New techniques have been developed in [LR10; CR11; Rob10; Cos11] for computing rational (ℓ,ℓ)-isogenies between abelian surfaces over finite fields.

We recall the following proposition from [CR11]:

Proposition 5.4.1 : *The complexity of computing an (ℓ,ℓ)-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 gives a way to compute the degree of the extension over which the points of a maximal isotropic kernel live.

Proposition 5.4.2 : *Let χ_π be the quartic polynomial satisfied by the Frobenius element for a smooth irreducible genus 2 curve C over $\mathbb{F}_q$ with simple, ordinary Jacobian $J(C)$. For a prime ℓ with $\ell \nmid q$, if there exists an $\mathbb{F}_q$-rational (ℓ,ℓ)-isogeny, then χ_π factors as $\chi_\pi = P\overline{P} \pmod{\ell}$ (where $\overline{P}$ is the conjugate of P under the action $\pi \to q/\pi$).*

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

Proof: Let $K \subset A[\ell]$ be a maximally isotropic rational kernel. Then since π stabilises K, if P is the characteristic polynomial of π restricted to K, then P divides χ_π. The cofactor is given by the characteristic polynomial of the action of the Verschiebung q/π on K, so it corresponds to $\overline{P}$. ■

5.5 Checking if the endomorphism ring is maximal

We recall the algorithm described in [EL10], and describe some improvements from [FL08; LR12]. The ideas for computing the endomorphism ring will also be used in the going up phase.

5.5.1 The vertical method

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

Proposition 5.5.1 : *Let $\mathcal{O} = \operatorname{End} A$ and let $\gamma \in \mathcal{O}_K$ be such that $\ell^e \gamma \in \mathbb{Z}[\pi,\overline{\pi}]$. There exists a unique integer polynomial P_γ of degree less than 4 such that $\ell^e p\gamma = P_\gamma(\pi)$, and γ is in $\mathcal{O}$ 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 [FL08, p. 38]), so that $\ell^e p\gamma \in \mathbb{Z}[\pi]$, which means we can write $\ell^e p\gamma = P_\gamma(\pi)$ for a unique $P_\gamma \in \mathbb{Z}[x]$ of degree less than 4. Second, since we are dealing with ordinary abelian surfaces over $\mathbb{F}_p$, we have $p \nmid [\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$ by [FL08, Proposition 3.7], so that

$\gamma \in \mathcal{O} \Leftrightarrow p\gamma \in \mathcal{O}$. Lastly, by the universal property of isogenies, we have that $P_\gamma(\pi) = 0$ on $A[\ell^e]$ if and only if $p\gamma \in \mathcal{O}$ (see [EL10]). Summing up, we only need to check that $P_\gamma(\pi) = 0$ on $A[\ell^e]$ to check that $\gamma \in \mathcal{O}$. ■

Remark 5.5.2 : Since most of the curves in the isogeny class are not maximal, it is more efficient to check the condition $P_\gamma(\pi) = 0$ on $A[\ell], A[\ell^2], \ldots$, rather than directly on $A[\ell^e]$. ♢

5.5.2 Reducing the degree

The obvious method of using Proposition 5.5.1 to test whether an element of $\mathcal{O}_K$ lies in $\mathcal{O}$ involves computing a basis of the ℓ^e-torsion group using §5.3.

The cost of such a computation depends on the degree of the extension where the ℓ^e-torsion points are defined. We have:

Lemma 5.5.3 : *Let d be the degree such that the ℓ-torsion points of A are defined over $\mathbb{F}_{p^d}$. Then $d \leqslant \ell^4 - 1$. Furthermore, the ℓ^e-torsion is all defined over an extension of degree d_e with $d_e = d\ell^{e-1}$.*

Proof: Let χ_π be the characteristic polynomial of π. Then d is the (multiplicative) order of X in the ring $\mathbb{F}_\ell[X]/\chi_\pi(X)$, so $d \leqslant \ell^4 - 1$. The second assertion follows from [FL08, Section 6]. ■

Remark 5.5.4 : For *maximal* abelian surfaces, [FL08, Proposition 6.2] gives a better bound for d: In that case we have $d < \ell^3$, and if ℓ is completely split in $\mathcal{O}_K$ we have $d \mid \ell - 1$. ♢

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 $(\pi^{d_0} - 1) \in \ell\mathcal{O}_K$. The proof of Lemma 5.5.3 show that d is the minimal integer such that $(\pi^d - 1) \in \ell\mathbb{Z}[\pi, \overline{\pi}]$.

In particular, $d_0 \mid d$, and, as remarked in [FL08], since we only need to check if $\mathcal{O} = \mathcal{O}_K$, we can first check that $(\pi^{d_0} - 1)/\ell$ lies in $\mathcal{O}$. In other words, we can check that the ℓ-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 ℓ^e-torsion points are then defined over an extension of degree $d_0\ell^{e-1}$ of $\mathbb{F}_p$, which allows us to work with smaller extensions.

One improvement 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 there are three possibilities: We have either $d_0' = d_0$, or $d_0' = 2d_0$, or $d_0 = 2d_0'$. In the third case it is to our advantage to replace A by its twist, because the Frobenius of the twist is represented by $-\pi$, and we can therefore compute the points of ℓ^e-torsion by working over extensions of half the degree.

Example 5.5.5 : Let H be the curve $y^2 = 80x^6 + 51x^5 + 49x^4 + 3x^3 + 34x^2 + 40x + 12$ of genus 2 over $\mathbb{F}_{139}$, and let J be the Jacobian of H. By computing the characteristic polynomial of Frobenius for J we find that

$$(\operatorname{End} J) \otimes \mathbb{Q} \cong \mathbb{Q}\left(i\sqrt{13 + 2\sqrt{29}}\right),$$

and we would like to check whether $\operatorname{End} J$ is maximal. In this example, we compute that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = 3^5$, so we need to compute the points in $J[3^5]$, which live 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.

5.5.3 Reducing the number of endomorphisms to test

One last improvement done in [LR12] is to use the fact that $\operatorname{End} A$ is an order; if we know that $\gamma \in \mathcal{O}$, then we know that the whole ring $\mathbb{Z}[\pi, \overline{\pi}, \gamma]$ is contained in $\mathcal{O}$. For example, suppose $\{1, \alpha_1, \alpha_2, \alpha_3\}$ is a basis for $\mathcal{O}_K$ and $\alpha_3 = \alpha_1 \alpha_2 \mod \mathbb{Z}[\pi, \overline{\pi}]^*$. To check that $\mathcal{O} = \mathcal{O}_K$ we only have to check that α_1 and α_2 are in $\mathcal{O}$. In fact, since the algorithm works locally at primes ℓ, we only need the relation between α_3 and $\alpha_1\alpha_2$ to hold locally at ℓ.

We use this idea as follows: Suppose that we have checked that $\{\gamma_1, \ldots, \gamma_k\}$ are endomorphisms lying in $\mathcal{O}$, and we want to check if $\gamma \in \mathcal{O}$. Let N_1 be the order of γ in the $\mathbb{Z}$-module $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}, \gamma_1, \ldots, \gamma_k]$, and N_2 be the order of γ in $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}]$. If we write $N_2 = \prod \ell_i^{e_i}$, we only have to check that $(N_2/\ell_i^{e_i})\gamma \in \mathcal{O}$ for $\ell_i \mid N_1$. In fact, if the valuation of N_1 at ℓ_i is f_i, then we would only need to check that $(N_1/\ell_i^{f_i})\gamma \in \mathcal{O}$, which means testing if $N_1\gamma = 0$ on the $\ell_i^{f_i}$-torsion, where $N_1\gamma$ is a polynomial in π, $\overline{\pi}$, and the γ_i $(i = 1, \ldots, k)$. We write this polynomial as $N_1/(pN_2)$ times a polynomial in π, so that we still need to compute the $\ell_i^{e_i}$-torsion.

Example 5.5.6 : Let H be the curve $y^2 = 10x^6 + 57x^5 + 18x^4 + 11x^3 + 38x^2 + 12x + 31$ of genus 2 over $\mathbb{F}_{59}$ and let J the Jacobian of H. We have

$$(\operatorname{End} J) \otimes \mathbb{Q} = \mathbb{Q}\left(i\sqrt{29 + 2\sqrt{29}}\right)$$

and we would like to check whether $\operatorname{End} J = \mathcal{O}_K$. The ring $\mathcal{O}_K$ is generated as a $\mathbb{Z}$-module by $1, \alpha, \beta, \gamma$, where α has order 2 in $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}]$, β has order 4, and γ has order 40. The algorithm from [FL08] would require computing the elements of $J[2^3]$ and $J[5]$. But $(\mathcal{O}_K)_2 = \mathbb{Z}_2[\pi, \overline{\pi}, \alpha]$, so we only need to compute in $J[2]$ and $J[5]$.

5.5.4 The algorithm

In final, we have the following algorithm:

Algorithm 5.5.7 : Checking that $\operatorname{End} A$ is maximal.

INPUT: An ordinary abelian surface $A/\mathbb{F}_p$ with CM by K.

OUTPUT: *True* or *false*, depending on whether or not $\operatorname{End} A = \mathcal{O}_K$.

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 ℓ-torsion lives.) For each $\ell \mid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ do:
    - (a) Let d be the smallest integer such that $\pi^d - 1 \in \ell\mathcal{O}_K$, and d' be the smallest integer such that $(-\pi)^{d'} - 1 \in \ell\mathcal{O}_K$. If $d' < d$, switch to the quadratic twist.
    - (b) Compute a basis of $A[\ell](\mathbb{F}_{p^d})$ using the algorithm from [BCR12].
    - (c) If this basis is of cardinality (strictly) less than 4, return *false*.
    - (d) (Checking the generators of $\mathcal{O}_K$.) For $i = 1, 2, 3$ do:
        - i. Let N_1 be the order of α_i in $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}, \alpha_j \mid j < i]$ and N_2 the order of α_i in $\mathcal{O}_K/\mathbb{Z}[\pi, \overline{\pi}]$.
        - ii. If $\ell \mid N_1$, let e be the ℓ-valuation of N_2 and write $pN_2\alpha_i$ as a polynomial $P(\pi)$.

iii. Compute a basis of $A(\mathbb{F}_{p^{d\ell^{e-1}}})[\ell^e]$.

iv. If $P(\pi) \neq 0$ on this basis, return *false*.

3) Return *true*. $\diamondsuit$

Remark 5.5.8 : Comments on the algorithm:

- With the way we choose the basis of $\mathcal{O}_K$, we have $e_1 \leqslant e_2 \leqslant e_3$ (for each ℓ 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 ℓ, 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 ℓ is such that $\ell^2 \nmid [\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$ 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 ℓ-torsion is defined over $\mathbb{F}_{p^{d_\ell}}$ (see [FL08]).
$\diamondsuit$

5.5.5 Complexity

We will measure complexity in terms of operations in the base field $\mathbb{F}_p$, and we will neglect factors of $\log(p)$. Since the index $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$ is bounded by a polynomial in p by [FL08, Proposition 6.2], evaluating the polynomials $P(\pi)$ (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 ℓ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$ where e is at most the ℓ-valuation of the index. According to Lemma 5.5.3 and Remark 5.5.4, the ℓ^e-torsion points live in an extension of degree at most $d = \ell^{e+3}$. Since $\#A(\mathbb{F}_{p^d}) = p^{2d(1+\varepsilon)}$, computing a random point in $A(\mathbb{F}_{p^d})[\ell^e]$ takes $\widetilde{O}(d^2)$ 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, using the naïve algorithm of simply computing all possible multiples. 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 Chapter 6 and 7):

Proposition 5.5.9 : *Let $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]] = \prod \ell_i^{e_i}$ be the decomposition of the index into powers of primes. Then checking if an abelian surface in the isogeny class is maximal can be done in time $\sum \widetilde{O}(\ell_i^{2e_i+6})$.*

5.6 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 to a maximal one via isogenies. This is not always possible and we will explain some of the obstructions. One difficulty was already illustrated in [BGL11, Example 8.2], where it was shown that there can be cycles in the isogeny graph involving only 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

(ℓ,ℓ) isogenies can be disconnected, and can even have isolated nodes. 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; furthermore, we cannot currently estimate the probability of failure.

As noted in [FL08], 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 still holds true if we just need to find a curve in the right isogeny class. Typically, we select p so that the probability of finding a curve in the right isogeny class is of magnitude $p^{3/2}$. This explain why we can afford to spend a lot of effort on going up from a curve in the right isogeny class.

If A is an ordinary abelian surface with CM by K, then for each ℓ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$, we try to find an (ℓ,ℓ)-isogeny path starting from A and going to A' such that $(\mathcal{O}_K)_\ell = (\operatorname{End} A')_\ell$. If this is possible, we let $A = A'$ in the next step (going to the next ℓ). 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 $\operatorname{End} A$ and the (ℓ,ℓ)-isogenous surfaces A', and keep A' if its endomorphism ring is bigger than that of A. In this section we will describe a more efficient algorithm, which combines the endomorphism ring checks of the preceding section with a going-up phase. Since we are working locally in ℓ, we may as well suppose that we are working over $\mathbb{Z}_\ell$.

5.6.1 Going up for one endomorphism

In this section, we suppose that we have an element $\alpha' \in \mathcal{O}_K$ such that $\alpha := \gamma \ell^e \alpha'$ lies in $\mathbb{Z}[\pi]$ for some $\gamma \in \mathcal{O}_K$ prime to ℓ. Starting from an abelian surface A in the isogeny class, we want to find an abelian surface A' such that $\alpha/\ell^e \in \operatorname{End} A'$ (or equivalently that $\alpha' \in \operatorname{End} A'$ locally at ℓ).

We saw in §5.5 that α/ℓ^e is in the endomorphism ring of A if and only if $\alpha(A[\ell^e]) = 0$, 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 α/ℓ^e being an element of $\operatorname{End} A$. The algorithm is as follows: For each (ℓ,ℓ)-isogenous surface A', we let $N' = \#\alpha(A'[\ell^e])$ and 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 surface in the right isogeny class.

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

Lemma 5.6.1 : *With notation and assumptions as above, let A' be an abelian surface isogenous to A such that $\#\alpha(A'[\ell^{\varepsilon+1}]) < \#\alpha(A[\ell^{\varepsilon+1}])$. Then the kernel of the isogeny $A \to A'$ intersects non-trivially with $k = \alpha(A[\ell^{\varepsilon+1}])$.*

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

This gives the following algorithm:

Algorithm 5.6.2 : Going up for one endomorphism α/ℓ^ε.

INPUT: An ordinary abelian surface $A/\mathbb{F}_p$ with CM by K, a prime power ℓ^e, and an $\alpha \in \ell^e \mathcal{O}_K$.

```
OUTPUT: An abelian surface $A'/\mathbb{F}_p$ isogenous to $A$ such that $\alpha/\ell^\varepsilon \in \mathrm{End} A'$, or fail.
    1) Set $\varepsilon = 1$.
    2) Compute $N(\varepsilon) = \#\alpha(A[\ell^\varepsilon])$.
    3) If $N(\varepsilon) = 1$, do:
        (a) If $\varepsilon = e$ then return $A$.
        (b) Otherwise, set $\varepsilon := \varepsilon + 1$, and go back to Step 2.
    4) At this point, $N(\varepsilon) > 1$. Let $\mathscr{L}$ be the list of all rational maximal isotropic subgroups of
       $A[\ell]$ which intersect non-trivially with $\alpha(A[\ell^\varepsilon])$. For $k \in \mathscr{L}$ do:
        (a) Compute $A' = A/k$.
        (b) Let $N'(\varepsilon) = \#\alpha(A'[\ell^\varepsilon])$.
        (c) If $N'(\varepsilon) < N(\varepsilon)$, set $A = A'$ and go back to Step 2.
    5) Return fail.                                                                    ◇
```

Remark 5.6.3 : As in §5.5 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 ℓ^ε-torsion points of A are defined over an extension of degree $d\ell^{\varepsilon-1}$. If moreover $(\pi^{d_0} - 1)/\ell \in \mathrm{End} A$ they are actually defined over an extension of degree $d_0\ell^{\varepsilon-1}$.

Therefore when we try to go up globally for all endomorphisms α, the first step is to try to go up for the endomorphism $(\pi^{d_0} - 1)/\ell$. During the algorithm, the obstruction N is given by the size of the kernel of $\pi^{d_0} - 1$, whose rank is 4 minus the rank of the ℓ-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. ◇

5.6.2 Going up globally

Let $\{1, \alpha_1/\ell^{e_1}, \alpha_2/\ell^{e_2}, \alpha_3/\ell^{e_3}\}$ be a generating set 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 surface A in the isogeny class, we want to find an abelian surface which is maximal at ℓ.

We could apply Algorithm 5.6.2 for each α_i/ℓ^{e_i}, but the algorithm does not guarantee that the endomorphisms already defined on A stay defined during the process, so we would observe loops on non-maximal abelian surfaces with this method. Moreover we want to reuse the computations of $A[\ell^\varepsilon]$, 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 α_i, we define N to be the global obstruction $N = \sum N_i$. We can then adapt the same method: For each (ℓ, ℓ)-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 ℓ, or until we are stuck, in which case we try to find a new random abelian surface in the right isogeny class.

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

The level ε of the individual obstruction we are working on depends on the endomorphism considered, so if we get stuck on level ε, we may have to look at level $\varepsilon+1$ even if not all endomorphisms $\alpha_i/\ell^\varepsilon$ are defined yet. For instance, in the case where we are only dealing with two generators, there are examples where $N_1(\varepsilon) = 1$, $N_2(\varepsilon) \neq 1$ and $N_1'(\varepsilon) = 1$, $N_2'(\varepsilon) = N_2(\varepsilon)$ for all (ℓ,ℓ)-isogenous abelian surfaces A', so we are stuck on level ε. However we can still find an isogenous A' such that $N_1'(\varepsilon+1) < N_1(\varepsilon+1)$.

Finally, as in Remark 5.6.3, 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 ℓ-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 5.6.4 : Going up.

INPUT: An ordinary abelian surface $A/\mathbb{F}_p$ with CM by K, and a prime ℓ.

OUTPUT: An abelian surface $A'/\mathbb{F}_p$ with $\operatorname{End} A = \mathcal{O}_K$ (locally at ℓ), or *fail*.

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

5.6.3 Cost of the going-up step

As in the genus 1 case, 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 5.6.6), we need to check that it will not dominate the complexity of the rest of the algorithm, so that in theory there will be no drawback to using it. Thus we need to estimate the cost of the going-up step.

[1] The degree of the extension where the full ℓ^ε-torsion is defined depends on whether Step 1 succeeded.

The going-up phase is a mix of endomorphism testing and isogeny computations. We already analysed the cost of the endomorphism testing in the preceding section. For the isogeny computation, the points in the kernel of rational (ℓ,ℓ)-isogenies live in an extension of degree at most ℓ^2-1. Transposing the analysis of §5.5.5 to this case shows that the computation of all of the points in these kernels takes at most $\widetilde{O}(\ell^4)$ operations in $\mathbb{F}_p$. There are at most $O(\ell^3)$ such kernels, and each isogeny computation takes at most $\widetilde{O}(\ell^4)$ operations in the extension. The final cost is at most $\widetilde{O}(\ell^9)$ operations in $\mathbb{F}_p$ for computing all isogenies. For each of the $O(\ell^3)$ isogenous abelian surfaces we do (part of) the endomorphism ring computation, which takes $\widetilde{O}(\ell^{2e+6})$ operations, according to §5.5.5. Since the global obstruction computed is of size $O(\ell^e)$, we do at most $O(e)$ steps. The global complexity is then given as follows:

Proposition 5.6.5 : *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 $\widetilde{O}(\sum \ell_i^{2e_i+9})$ operations in the base field.*

Remark 5.6.6 : It is important to note that the going-up phase does not always succeed. First, as noted in the introduction of this section, the (ℓ,ℓ)-isogeny graph is not always connected, so if we start with a curve not in the same component as a maximal curve, there is no way to find the maximal curves using only (ℓ,ℓ)-isogenies. Second, even if the curve is in the same component as a maximal curve, finding a maximal curve may involve going through isogenous curves that increase 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 (ℓ,ℓ)-isogenies for any curve in the isogeny class. If χ_π is the characteristic polynomial, this can be detected by the fact that χ_π does not factor modulo ℓ 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 ℓ. This gives a criterion for estimating whether one can go up for this ℓ. ◊

5.7 Complexity of finding a maximal curve

Let p be a CRT prime. In this section, we discuss the expected complexity of finding a curve with CM by $\mathcal{O}_K$ using the tools from above. To fix the ideas, both in the CRT and p-adic method, p will be of size $\widetilde{O}(\Delta_0\Delta_1)$. Furthermore, both methods allow to select the prime p such that the isogeny class corresponding to π is relatively large (see §7.2) and that the going-up step and endomorphisms rings computations are negligible (see §7.3).

In particular, to have a quasi-linear algorithm to compute the class polynomials with these methods, we need to at least have a quasi-linear algorithm in the size of p to find such a maximal curve (and we need even more for the CRT method because we need to repeat this step $\widetilde{O}(\sqrt{\Delta_0\Delta_1})$ times.

We will see that the problem of genus 2 is that unfortunately the size of the isogeny class is too small compared to the size of all genus 2 curves to achieve quasi-linearity. More precisely, we expect to have $\Omega(p^{3/2})$ curves in the isogeny class for $O(p^3)$ isomorphism class of curves of genus 2. So just to find a curve in the right isogeny class takes time $O(p^{3/2})$.

If X is the number of going-up steps we then need to try on average, the cost to find a maximal curve is then expected to be $\widetilde{O}(X(p^{3/2}))$. At best, $X=O(1)$, while at worst $X=O(p)$ (number of random tries in the isogeny class until we find a maximal one directly).

5.8 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 (either over $\mathbb{F}_p$ for the CRT method, or over $\mathbb{Q}_p$ for the p-adic method).

As noted in Chapter 2, the set of maximal curves in the isogeny class corresponding to a fixed CM-type Φ is a principal homogeneous space under the action of the Shimura class group $\mathfrak{C}$ associated to the primitive quartic CM field K, which acts by isogenies.

However, using the Magma package AVIsogenies [BCR12] we can only compute isogenies with a maximal isotropic kernel. The lemma below show that in terms of the Shimura class group, this means that we can only compute the action corresponding to (equivalences classes) of elements of the form (I,ℓ), where I is an ideal in K and ℓ is a prime number.

Lemma 5.8.1 : *Let (I,ρ) be an element of the Shimura class group $\mathfrak{C}$ and let ℓ be a prime. Then the action of (I,ρ) on a maximal abelian surface 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 ℓ).*

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

We can prove that using these isogenies is enough to compute the action of the image of the type norm.

Proposition 5.8.2 : *There is a polynomial P such that for every primitive quartic CM field K, the image of the type norm in the Shimura class group associated to K is generated by elements of the form (I,ℓ), where ℓ ranges over the prime numbers less than $P(\log\Delta)$ and where Δ is the discriminant of K.*

Proof : Under GRH, we know that the class group of the reflex field is generated by prime ideals of degree 1 and of norm polynomial in $12\log\Delta'$ [Bac90, Theorem 1] where Δ' is the discriminant of the reflex field. But if I is such an ideal of $\mathcal{O}_{K^r}$ of norm prime to p, then the element $(\mathrm{TN}(I),N(I))$ will give a horizontal isogeny whose kernel is maximally isotropic for the ℓ-torsion.

Now it suffices to remark that Δ' is at worst in $O(\Delta^2)$ to conclude. ■

Unfortunately, while this is sufficient for the p-adic method, for the CRT method we can't compute irreducible component, so we need to compute the whole action of the Shimura class group. Therefore to ensure that we can find all other maximal curves using isogenies with maximal isotropic kernel, we make the following heuristic assumption.

Assumption : There is a polynomial P such that for every primitive quartic CM field K, the Shimura class group associated to K is generated by elements of the form (I,ℓ), where ℓ ranges over the prime numbers less than $P(\log\Delta)$ and where Δ is the discriminant of K. ◊

Proof (Justification) : We have tested this assumption on numerous examples, using the same bound as in Proposition 5.8.2. ■

Remark 5.8.3 : With the CRT method, in the horizontal step, by Proposition 5.8.2 we can compute the action of $\mathrm{TN}(\mathrm{Cl}(\mathcal{O}_{K^r}))$ by isogenies of size logarithmic in Δ. By Lemma 6.5 of [BGL11], 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 [BGL11, p. 516], we have $w(n) < 2\log\log n$ outside a density-0 subset of very smooth integers, so the corresponding factor can be absorbed into the $\widetilde{O}$-notation of the complexity analysis made in Chapter 7. ♢

It can be hard to associate an isogeny to a given element of the Shimura class group. However, is the degree of the isogeny is prime to the index $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$, then this isogeny has to be horizontal, or in other words come from the action of the Shimura class group.

Lemma 5.8.4 : *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 factorises through f, so $d = f\widetilde{f}$ for some isogeny $\widetilde{f} : B \to A$. By assumption d is prime to the index. If $\alpha \in \mathrm{End} A$, then $f \circ \alpha \circ \widetilde{f} = d\alpha$ 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 $\mathrm{End} B \subseteq \mathrm{End} A$, 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 ℓ as small as possible, since the size of ℓ 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 ℓ, then not all new (ℓ,ℓ)-isogenous abelian surfaces will be maximal, so we have to use Algorithm 5.5.7 to test which of them is maximal. In that case, after the isogeny is applied, the ℓ^e-torsion (in the notation of §5.6) must again be computed, 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 ℓ^e-torsion for small ℓ dividing the index versus the degree of the field of definition for the points in the kernel of the ℓ-isogeny for ℓ 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 $\alpha_1, \dots, \alpha_t \in \mathfrak{C}$, and we want to find new maximal curves by computing (ℓ,ℓ)-isogeny graphs starting from these curves. Then if $\mathfrak{C}(\ell)$ is the set of elements of the form (I,ℓ) 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 α_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 ℓ divides the index, because then we avoid 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 5.8.5 : Finding all maximal curves from one maximal curve.

INPUT: An ordinary abelian surface A/k with CM by $(\mathcal{O}_K, \Phi)$.

OUTPUT: All abelian surfaces over k with CM by $(\mathcal{O}_K, \Phi)$.

1) Precomputation: Compute a set of generators of the Shimura class group with relative norm ℓ 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.

2) For each generator of (relative) norm ℓ dividing the index, replace the previous degree by the degree of the extension where the ℓ^e-torsion lives. (Usually e is the ℓ-valuation of the index, but the tricks from §5.5 can sometimes reduce it.)

3) Sort the generators by the corresponding degrees to get a list $(g_1, \dots, g_n)$.

4) For each generator g_i on the list, let ℓ_i be its norm and do:

   (a) Compute the surfaces (ℓ_i, ℓ_i)-isogenous to the one already found. If ℓ_i divides the index, then do an endomorphism ring computation from §5.5 and keep only the maximal curves.

   (b) Repeat until the number of maximal abelian surfaces equals $|\langle \mathfrak{C}(\ell_1), \dots, \mathfrak{C}(\ell_i)\rangle|$. ♢

5.8.1 Complexity

For the horizontal step, the isogeny computation involves primes of size logarithmic in Δ, so the cost of this step is quasi-linear in the number $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ of maximal curves times the cost of an operation in the field we work with.

For the CRT method, this is under the Assumption above, but see Remark 5.8.3.

5.9 Perspectives

The cost of the endomorphism ring computation depends on the size of the prime powers dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. Gaëtan Bisson developed in [Bis11] a subexponential algorithm to compute the endomorphism ring of an abelian surface, extending previous work in genus 1 by him and Andrew Sutherland [BS09]. We briefly recall how this method work: If $\mathcal{O}_1$ is a suborder of $\mathcal{O}_2$ one can find relations in the class group of $\mathcal{O}_2$ that are not relations in the class group of $\mathcal{O}_1$. One can also find such relations when replacing the class groups by Shimura class groups, see [BS13]. Then if one starts with an abelian variety A with order either $\mathcal{O}_1$ or $\mathcal{O}_2$, and one follows horizontal isogenies given by these relations, either one goes back to A, in which case the endomorphism ring has to be $\mathcal{O}_2$, or one does not loop back to A, in which case the endomorphism ring has to be $\mathcal{O}_1$.

Since we still need to take ℓ-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. Even if this is not the case, for the going-up step we have to work with extensions of degree at most ℓ^2 (the degree of the extensions where the geometric points of rational isogenies live), while computing the action of endomorphisms on the full ℓ-torsion can requires to go to a degree up to ℓ^4.

Gaetan Bisson, Kristin Lauter and Damien Robert have developed a yet unpublished method that combines the going-up algorithm with the endomorphism ring algorithm. Roughly speaking, with the notations from above, in the absence of a loop, we know that we have CM by $\mathcal{O}_1$. Suppose that the index of $\mathcal{O}_2$ in $\mathcal{O}_1$ is a prime ℓ, and that we want to find an abelian variety with CM by $\mathcal{O}_2$. If B is the end point of the isogeny path, we check for two ℓ-isogenies, one starting from A and one starting from B, that go to the same end point C. Then, under some assumptions, C has indeed CM by $\mathcal{O}_2$.

Once we have a maximal curve, we have seen in §5.6 that if a prime ℓ divides the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, it can be hard to know whether we have an horizontal isogeny or a vertical one. In [Ion12], Sorina Ionica gave a method based on the Tate pairing (in the same spirit as [IJ10]) of the points of the kernel of the isogeny to determine in advance in which case we are.

We have seen in §5.6 that the going-up method may fail, because we are limited to isogenies with maximal isotropic kernel. Recent progress about using real multiplication to compute isogenies with cyclic kernels [Rob13] give hope about always being able to go up. In this context, there is work by Sorina Ionica to extend her algorithm in [Ion12] in order to be able to determine if a cyclic isogeny will go up according to the value of the Tate pairing of its generator (and not just determine if an isogeny will stay maximal once we are on a maximal curve).

But we have seen in §5.7 that even finding a curve in the right isogeny class is too expensive. One explanation for this is that we try to compute the class polynomials (describing a scheme of dimension 0) directly from the moduli space of dimension 3 of all abelian surfaces. By 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, or Gundlach invariants, as proposed in [LY11]. Heuristically, among the $p^{3/2}$ curves in the isogeny class, we would expect $\Omega(p)$ to have maximal real multiplication $\mathcal{O}_{K_0}$. Since the Hilbert moduli space for $\mathcal{O}_{K_0}$ is of dimension 2, we would expect to find a curve in the right isogeny class by working inside the Humbert surface in time quasi-linear in the size of p.

Of course, this would require an algorithm to obtain the equations of the Humbert surface. Potentially, all three methods could be adapted to obtain such equations (see [Gru10]).

6 The p-adic method

6.1 Overview

We briefly recall how the p-adic method works.

Algorithm 6.1.1 :

INPUT: A primitive quartic CM field K with a CM-type Φ, and a CRT prime p for K.

OUTPUT: Igusa class polynomials $H_i(X)$, $i = 1, 2, 3$, in $K_0^r[X]$.

1) Find a curve with CM by $\mathcal{O}_K$ over $\mathbb{F}_p$.
2) Lift the invariants of this curve to $\mathbb{Q}_p$ up to a certain precision.
3) Use horizontal isogenies coming from the action of the type norm over $\mathbb{Q}_p$ to recover the other maximal curves over $\mathbb{Q}_p$.
4) Use LLL to recover the Igusa class polynomials in $K_0^r[X]$ from the invariants. ◇

We note that since p is a CRT prime, p splits completely in K_0^r, and if $\mathfrak{p}$ is a prime above p, then the completion of K_0^r at $\mathfrak{p}$ is isomorphic to $\mathbb{Q}_p$. Hence it makes sense to recover coefficients of K_0^r inside $\mathbb{Q}_p$.

We have already seen in Chapter 5 most of the steps of these algorithm. We also refer to Chapter 7 which uses very similar tools.

The only new step is the lifting, which we describe now.

6.2 Computing the canonical lift of an abelian surface

By definition, the canonical lift preserves the endomorphism ring, and by functoriality the isogenies. To compute the canonical lift of A, a standard method is to find a cycle of isogenies from A, and to carry out multivariate Newton iterations to lift this cycle. This typically yields a method to compute a canonical lift in a quasi-linear time in the precision.

For instance, one can use the path given by the Frobenius (or the Verschiebung). With elliptic curves, lifting the corresponding isogeny path amounts to lifting solutions for the modular polynomial of degree p.

6.2.1 Characteristic 2

When $p = 2$, one can use the arithmetic-geometric mean (AGM, in genus 1) or Borchardt mean (in genus 2) to lift an abelian surface defined over an extension $\mathbb{F}_{2^d}$. The interpretation of the Borchardt mean as coming from the duplication formulæ on ϑ-constants (see §4.2) show that the it yields a cycle of 2-isogenies, so it fits in the preceding framework.

This idea was suggested by Mestre in [Mes02]. Rather than working directly with ϑ-constants, in [GHK+06] the authors use the Richelot correspondence, which is the geometric realisation of the Borchardt mean on hyperelliptic curves.

It is remarkable that the Borchardt mean, which appears in the analytic method, can also be used p-adically.

6.2.2 Characteristic $p > 2$

In [CKL08], the authors develop a degree 3 correspondence on ϑ-constants in order to lift an abelian surface over $\mathbb{F}_{3^d}$. This method has been extended to all characteristics in [CL09] (although the authors do not mention the applications to class field computation). We note that the initialising step of this last algorithm uses an expensive Gröbner basis algorithm. This initialising step has been (implicitly) improved in [FLR11; LR10].

Of course, one could also use the action of the Shimura class group to compute cycles and lift these cycles (by lifting the kernel of the isogenies). This would have the advantage of already doing part of the job of the horizontal isogenies in $\mathbb{Q}_p$.

6.3 Complexity

In [GHK+06], the authors start with an abelian curve with CM by $\mathcal{O}_K$ over the field $\mathbb{F}_{2^d}$ with d "small". They use Richelot isogenies to lift in time quasi-linear in the precision P (which is $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$). However, since at the time they did not have access to isogenies in dimension 2, to recover the class polynomials from the lifted invariant they use an LLL algorithm to compute its minimal polynomial. They find a complexity in $\widetilde{O}(D^5P)$ where $D = \widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ is the degree of the class polynomials. In total this gives a quasi-cubic algorithm in the size of the class polynomials.

Nowadays, we can use isogenies to compute the conjugates directly, see §5.8). We need D isogenies of logarithmic degree working at precision P, for a quasi-linear cost.

However, what is hidden in [GHK+06] is the cost of finding a CM curve over $\mathbb{F}_p$. By the analysis done in §5.7, this is actually the dominant step and is not quasi-linear (except if the ideas of §5.9 prove fruitful)!

7 The CRT method

7.1 Overview

We recall how the CRT method work:

Algorithm 7.1.1 :

```
INPUT: A primitive quartic CM field $K$ with a CM-type $\Phi$, and a collection of CRT primes $P_K$ for $K$.
OUTPUT: Igusa class polynomials $H_i(x)$, $i = 1,2,3$, either in $K_0^r[x]$ or reduced modulo a prime $\mathfrak{q}$.
    1) Loop through CRT primes $p \in P_K$:
        (a) 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.
        (b) Try to go up to a maximal curve from $C$; if this step fails, go back to Step 1(a).
        (c) From a maximal curve $C$, compute all other maximal curves.
        (d) Reconstruct the class polynomials $H_i(x)$ modulo $p$ from the Igusa invariants of the set of maximal curves.
    2) Recover $H_i(x)$, $i = 1,2,3$ in $K^r[x]$ or modulo $\mathfrak{q}$ using the (explicit) CRT method once we have computed $H_i(x)$ modulo $p$ for enough primes $p$.
```
♢

The CRT algorithm terminates when the lifted class polynomials (with denominators recovered via LLL) are constant from one CRT prime to the next. The probability that the class polynomials are correct when this happens was estimated in [FL08, Remark 7.2].

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) [Str10, Chapter 3]. It is easy to compute these orbits modulo each CRT prime p using the tools from Chapter 5. However we need to be able to glue the correct orbits together when doing the CRT. For this, one possible way is to use the "trace trick" from [ES10b]. In this method, the trace of the class polynomials is computed (for instance via the analytic method), and we use it to glue correctly the irreducible component across several CRT primes. This "trick" only work because the trace is in practice much smaller than the other coefficients.

7.2 Strategies for sieving CRT primes p

Since we have some latitude in the CRT primes p, we can sieve the primes 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.

7.2.1 Cost of testing if a curve is maximal

Before using a prime for the CRT computation, we first need to check whether testing if a curve is maximal is too expensive. For this, we compute which subgroups $A[\ell^e]$ are needed to compute to test if A is maximal as in §5.5, and in which extensions the points of these subgroups are defined. As already remarked, for ℓ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, e is usually the valuation of ℓ in the index, but some of the tricks from §5.5 can reduce it.

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

7.2.2 Size of the isogeny class

For each CRT prime p, the first phase of the algorithm relies on 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 [LPP02] 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 π can be unfortunately small. In those cases, the 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 sizes of the Shimura class groups associated to all the orders containing $\mathbb{Z}[\pi, \overline{\pi}]$ which are stable under complex conjugation. Of course we do not compute the Shimura class groups in order to make this estimate (even computing only the class group can be too expensive for suborders of large discriminant), but we need to estimate their sizes.

Lemma 6.3 in [LPP02] 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 $\mathbb{Z}[\pi, \overline{\pi}]$ and this is already too expensive. In practice, it is enough to estimate the size of the isogeny class using only the factorisation 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 [LPP02]

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

where c is the size of the co-kernel of the norm map 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 [LPP02] gives us

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

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 ℓ^e is a prime power dividing I, and $\chi_\pi = \prod P_i^{e_i}$ the factorisation 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 ℓ in $\mathcal{O}_K$.

Now we use the following estimate: for each divisor d of the index I, the contribution of orders $\mathcal{O}$ such that $[\mathcal{O} : \mathbb{Z}[\pi,\overline{\pi}]] = d$ to the estimate of the number of 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 an ℓ where we can't go up). We refer to [LR12] to how this estimate compares to some real examples.

7.2.3 Estimating the probability of going up

In practice, we are not interested in the size of the isogeny class, but in the number of curves in the isogeny class from which we can go up. From numerous experiments, we have observed that most of the cases where we can't go up for a particular ℓ arise because no rational (ℓ,ℓ)-isogenies exist at all. But we can easily detect this case by using Proposition 5.4.1.

We can thus estimate the number $\mathbf{C}$ of curves from which 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 ℓ where we cannot 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 succeeding. Now we keep the prime p if $\mathbf{C}_0$ is not too small, and if the cost of doing all these endomorphism ring computations 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).

7.2.4 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 number of curves $\mathbf{C}$ 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 for 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 previously discarded CRT primes, and we use the prime giving the maximum probability provided it is more than $1/16p^{-3/2}$.

7.3 Complexity

In this section, we give a mostly heuristic analysis of how Algorithm 5.6.4 (the going-up algorithm) and Algorithm 5.8.5 (the algorithm to find all maximal curves from one maximal curve) affect the asymptotic complexity of Algorithm 7.1.1.

Recall that the degree of the class polynomials is $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$, while we use the observed bound of $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ for the coefficient height.

According to [BGL11, §6.4], the smallest CRT prime is of size $\widetilde{O}(\Delta_0\Delta_1)$. We need $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ CRT primes, and an analysis using [LO77], as in [BBE+08, §5, Lemma 3], shows that the largest prime is also $\widetilde{O}(\Delta_0\Delta_1)$. 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.

In contrast, 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 [FL08] we have that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] \leqslant 16p^2/\sqrt{\Delta}$. For the size of the CRT prime we are considering, we see that $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = \widetilde{O}(\Delta_0\Delta_1^{3/2})$. We fix $\varepsilon = 1/2$. Assuming that the index is uniformly distributed, [Dic30] showed that there is a positive density of CRT primes where the largest prime power dividing the index is $O(\Delta_0^{\varepsilon/100}\Delta_1^{\varepsilon/100})$. By the complexity analysis of §§5.5.5 and 5.6.3, we see then that there is a positive density of primes where these algorithms take time at most $O(\Delta_0^{\varepsilon}\Delta_1^{\varepsilon})$. This justifies the assumption made in §5.7.

We then let $p = \widetilde{O}(\Delta_0\Delta_1)$ 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 [BGL11, Heuristic 6.6]. Up to isomorphism over the algebraic closure, there are p^3 genus-2 curves over $\mathbb{F}_p$. The original CRT algorithm of [EL10; FL08] looped through all p^3 geometric isomorphism classes of curves and tested whether the corresponding endomorphism ring is maximal. This takes time $\widetilde{O}(\Delta_0^3\Delta_1^3) + O(\Delta_0^{3/2+\varepsilon}\Delta_1^{3/2+\varepsilon})$ per CRT prime. Since $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ CRT primes are needed, we find a total cost of $\widetilde{O}(\Delta_0^{7/2}\Delta_1^{7/2})$ given our choice of ε.

Of course, once a maximal curve is found, we can use horizontal isogenies to find the others as explained in Chapter 5. This approach was suggested in [BGL11] and fully developed in [LR12]. It yields a cost of $\widetilde{O}(\Delta_0^{5/2}\Delta_1^{5/2}) + O(\Delta_0^{3/2+\varepsilon}\Delta_1^{3/2+\varepsilon})$ per CRT prime. The total cost is then $\widetilde{O}(\Delta_0^3\Delta_1^3)$.

Lastl by summing up the going-up steps, if X is the number of going-up steps we need to try on average, the cost per CRT prime is $\widetilde{O}(X(\Delta_0^{3/2}\Delta_1^{3/2} + \Delta_0^{\varepsilon}\Delta_1^{\varepsilon}))$. At best, $X = O(1)$, and we have a total cost of $\widetilde{O}(\Delta_0^2\Delta_1^2)$ from CRT primes. So at best we have a quasi-quadratic complexity, while the CRT itself is quasi-linear, and thus negligible. We see that we are still far from quasi-linearity as 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.

We see that the step described in §5.7 dominates the complexity of the CRT method. So to improve its complexity, one would need to implement some of the methods suggested in §5.9. Even with the quasi-linear method to find a curve with CM by $\mathcal{O}_K$ in $\mathbb{F}_p$ suggested at the end of this section, since we need to use $\widetilde{O}(\Delta_0^{1/2}\Delta_1^{1/2})$ CRT primes, we would still get a cost of $\widetilde{O}(\Delta_0^{3/2}\Delta_1^{3/2})$, which would still not be quasi-linear.

8 Implementation and examples

Currently, the analytic method in genus 2 is the fastest one available, so we concentrate on this approach in the following.

8.1 Implementation and parallelisation

The implementation by A. Enge and E. Thomé of the quasi-linear complex-analytic algorithms will soon be available at

http://cmh.gforge.inria.fr/.

The software implements the different steps of Algorithm 4.1.1 as follows:

- Steps 1) to 3) of Algorithm 4.1.1 are performed by a script in PARI/GP[Bel12], which does all computations symbolically, and the running time of which is essentially negligible.

- The computation of ϑ-constants in Step 4) of Algorithm 4.1.1 is done by a C program, based on the library GNU MPC[EGT+12], itself using the GNU MPFR[HLP+12] and GNU MP[Gra13] libraries. Newton lifting is used for this step from a base precision of 2000 bits, and it is parallelised through MPI.

- Reconstruction of the class polynomials from the numerical values of the Igusa invariants is done inside the same C program, relying on the library MPFRCX[Eng12] for basic operations on polynomials using the FFT and asymptotically fast algorithms on trees of polynomials. In a preparatory step, the leaves of the tree for H_1 are filled with the linear factors of the class polynomial, those for $\widehat{H}_k$, $k = 2, 3$, are filled with the values of j_k. Let the subscripts l and r denote the left and the right descendant, respectively, of a given node. Then an inner node $n^{(1)}$ in the tree for H_1 is computed as $n^{(1)} = n_{\mathrm{l}}^{(1)} \cdot n_{\mathrm{r}}^{(1)}$, while an inner node $n^{(k)}$ in the tree for $\widehat{H}_k$, $k = 2, 3$, is obtained as $n_{\mathrm{l}}^{(k)} \cdot n_{\mathrm{r}}^{(1)} + n_{\mathrm{l}}^{(1)} \cdot n_{\mathrm{r}}^{(k)}$, where $n^{(1)}$ denotes the node at the same position in the tree for H_1; for details, see [GG99, Algorithms 10.3 and 10.9]. By first combining pairs of complex-conjugate leaves in a preprocessing step, all computations are in fact carried out with real floating point polynomials, see [EM03]. So if at a given level the tree for H_1 contains m nodes, all nodes at this level of the three trees can be obtained with $5m$ independent multiplications, which are parallelised using MPI.

- Recognition of the polynomial coefficients as elements in K_0^r is also embedded in the C program, using FPLLL[CPS13] for the LLL step.

- Validation of the obtained class polynomials is performed by computing a Weil number π above a prime $p \approx 2^{128}$, constructing a curve over $\mathbb{F}_p$ having as endomorphism ring the ring of integers of K using Mestre's algorithm [Mes91], and verifying that the cardinality of the Jacobian matches $\mathrm{N}_{K/\mathbb{Q}}(1 \pm \pi)$. This step is done in PARI/GP and also has a negligible cost.

In the following we report on the performance of these different steps, illustrated by both small and large examples.

8.1.1 Computation of ϑ-constants

We report timing results for the computation of fundamental ϑ-constants for two arbitrary period matrices. Table 8.1 shows that already our implementation of the relatively simple naive algorithm presented in §4.2.1 may be several orders of magnitude faster[1] than MAGMA-2.19.4, the performance improvement ratio depending on the period matrix. Newton lifting is preferable above some cut-off value for the precision, here 16 000 and 4 000 bits, respectively. The naive algorithm is rather sensitive to the period matrix; generally speaking, it converges the faster the larger the imaginary parts in Ω are, which correspond to smaller q_0, q_1, q_2. A noticeable difference between our naive algorithm from §4.2.1 and the implementation in MAGMA is that the favorable cases are not the same. This is most likely due do different choices of summation regions, as briefly discussed in §4.2.1. We note that the timings of Newton lifting depend much less on the concrete period matrix entries than those for the naive method.

	$\Omega = \begin{pmatrix} \frac{-1+5i}{2} & \frac{i}{6} \\ \frac{i}{6} & \frac{-1+7i}{2} \end{pmatrix}$			$\Omega = \begin{pmatrix} \frac{2+10i}{7} & \frac{1+2i}{6} \\ \frac{1+2i}{6} & \frac{4}{10}+8i \end{pmatrix}$		
bits	MAGMA	CMH-naive	CMH-Newton	MAGMA	CMH-naive	CMH-Newton
$\approx 2^{11}$	0.46	0	0.02	0.03	0	0.02
$\approx 2^{12}$	3.4	0.01	0.04	0.17	0.04	0.03
$\approx 2^{13}$	26	0.07	0.08	1.1	0.20	0.09
$\approx 2^{14}$	210	0.31	0.24	8.2	1.0	0.26
$\approx 2^{15}$	1700	1.3	0.69	60	5.2	0.75
$\approx 2^{16}$		6.4	2.0	430	27	2.2
$\approx 2^{17}$		32	5.7	3100	130	6.0
$\approx 2^{18}$		160	16		720	16
$\approx 2^{19}$		770	39		3100	40
$\approx 2^{20}$		3200	98			96
$\approx 2^{21}$			240			230
$\approx 2^{22}$			560			530
$\approx 2^{23}$			1400			1300
$\approx 2^{24}$			3200			3000
$\approx 2^{25}$			7600			7100
$\approx 2^{26}$			16000			16000

Table 8.1: Calculation of $\vartheta_0(\tau)$ (Intel i5-2500, 3.3GHz; MAGMA-2.19.4; CMH-1.0).

Notice that the running times for Newton lifts are consistent with the theoretical complexity of $O(\mathsf{M}(N)\log N)$. The code in CMH implements the approach using finite differences for estimating the Jacobian matrix as described in §4.2.4, as well as an algorithm which computes the exact Jacobian matrix along with the Borchardt mean as given in [Dup06, Algorithme 16]. Both converge equally well, but the latter approach is computationally more expensive by roughly 45 %, accounted for by a larger number of multiplications.

8.1.2 Breakdown of timings for small class polynomial examples

Table 8.2 illustrates our class polynomial computations on relatively small examples.

Our code distinguishes orbits of the roots of the Igusa class polynomials under complex conjugation. For instance, there are four real roots and 58 pairs of complex-conjugate roots in the second example, so

[1] Such a quadratic, yet efficient implementation was used by T. Houtmann to compute class polynomials of degree up to 500 (personal communication, no reference exists).

$K = \mathbb{Q}[X]/(X^4 + 144X^2 + 3500)$
$\mathfrak{C} = \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/30\mathbb{Z}$

preparation	0.2
base, 2 000 bits	0.6
lift, 3 984 bits	0.8
lift, 7 944 bits	2.1
reconstruction attempt	0.1
lift, 15 846 bits	6.2
$H_1, \widehat{H}_2, \widehat{H}_3 \in \mathbb{C}[X]$	0.1
$H_1, \widehat{H}_2, \widehat{H}_3 \in K_0^r[X]$	3×0.3
check	0.8
Total (incl. I/O)	12.4

$K = \mathbb{Q}[X]/(X^4 + 134X^2 + 712)$
$\mathfrak{C} = \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/60\mathbb{Z}$

preparation	0.3
base, 2 000 bits	1.1
lift, 3 988 bits	1.6
lift, 7 958 bits	4.4
reconstruction attempt	0.1
lift, 15 886 bits	13.1
reconstruction attempt	0.2
lift, 31 744 bits	38.7
$H_1, \widehat{H}_2, \widehat{H}_3 \in \mathbb{C}[X]$	0.6
$H_1, \widehat{H}_2, \widehat{H}_3 \in K_0^r[X]$	1.8 + 2×1.4
check	0.7
Total (incl. I/O)	69.2

Table 8.2: Timings in seconds for two examples (on one Intel i5-2500, 3.3GHz

that altogether we need to carry out 62 lifts of ϑ-constants. Instead of targeting a given precision based on arguments as developed in [Str09], we simply carry out successive lifting steps until the polynomial reconstruction succeeds. This explains the time needed for failed reconstruction attempts in Table 8.2, which could be avoided if we had a sharper bound on the required precisions. It regularly occurs, even though this is not illustrated by the examples here, that the reconstruction of the class polynomial $H_1 \in K_0^r[X]$ succeeds one lifting step before that of $\widehat{H}_2, \widehat{H}_3 \in K_0^r[X]$. This can be explained by the relative size of the invariants considered by Streng, see [Str10, Appendix 3].

The timings indicated as "preparation" and "check" in Table 8.2 correspond to the number theoretic calculations performed in PARI/GP. The preparation time covers the enumeration of $\mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) \subseteq \mathfrak{C}$, and the creation of the relevant set of reduced period matrices. Checking means finding a Weil number over a 128-bit prime and generating a genus 2 curve the Jacobian of which has complex multiplication by the maximal order of K.

8.2 A large example

Our currently largest example is $K = \mathbb{Q}[X]/(X^4 + 1357X^2 + 2122)$, containing $K_0 = \mathbb{Q}(\sqrt{1832961})$ of class number 8. Its Shimura class group is $\mathfrak{C} = \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) \simeq \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/4402\mathbb{Z}$ of size 17608. On one core of an Intel Core i7-2620M clocked at 2.7 GHz, the structure of the class group is obtained with our PARI/GP script in only 100 ms, while the computation of the period matrices and their symbolic reduction into the fundamental domain $\mathscr{F}_2$ takes 230 s.

The associated ϑ-constants consist of 8804 pairs of complex-conjugate values. For the first eleven Newton iterations up to a precision of about 4 000 000 bits, we used 640 cores Intel Xeon X5675 at 3.07 GHz; for the last iteration, we switched to a machine with only 160 cores Intel Xeon E7-8837 at 2.67 GHz, but with 640 GB of main memory. Table 8.3 gives the timings (in seconds) for the Newton lifts of one particular period matrix. The small value of δ, estimated as explained at the end of §4.2.4, and which quickly stabilises at a more or less constant value as predicted by Theorem 4.2.10, shows that the effective precision indeed almost doubles in each step.

The lifting step accounts for a total of about 510 CPU days, but thanks to its easy parallelisation on

precision	δ	time
2 000	—	0.04
3 990	10	0.1
7 958	22	0.2
15 882	34	0.7
31 724	40	1.8
63 408	40	5.2
126 772	44	14
253 504	40	42
506 966	42	99
1 013 890	42	220
2 027 738	52	510
4 055 434	22	1 100
8 110 826	42	3 000

Table 8.3: Time for lifting steps for example with $\#\mathfrak{C} = 17\,608$.

160 to 640 cores, it was finished in less than 3 days wall-clock time.

The computation of the floating point polynomials H_1, $\widehat{H}_2$ and $\widehat{H}_3$ was carried out at a precision of 7 536 929 bits (the lowest lifting precision reached for one of the period matrices). The first step consists of $5 \cdot 8804/2 = 22010$ multiplications of monic polynomials of degree 2, which can be arbitrarily parallelised; we used again the machine with 160 cores and 640 GB of memory. From degree 1 024 on, we switched to a machine with 40 Intel Xeon E7-4870 cores at 2.4 GHz and 1 TB of memory. In degree 4 096, this allowed to use one Karatsuba step, replacing the 10 multiplications by 30 multiplications of half the degree carried out in parallel. In degree 8 192, this was not possible due to the amount of memory used in the underlying FFT multiplications. In the last step, we needed to multiply a degree 16 384 polynomial with a degree 1 224 degree polynomial. Using 3-way Toom–Cook, we could replace the 5 multiplications by 25 multiplications of size 3 times smaller. The wall-clock time of this polynomial reconstruction step was about 1 day.

Recognising one coefficient of the floating point polynomials as an element of K_0^r took about 2 000 s per coefficient on one Intel Xeon X5675 core at 3.07 GHz. The total CPU time for the 52 825 coefficients was thus about 1 200 CPU days; with up to 960 cores working in parallel, this took less than 2 wall-clock days.

The uncompressed storage size of the three resulting polynomials in base 10 is about 56 GB. The common denominator of the coefficients of H_1 has 3 465 distinct prime factors, the largest one being 242 363 767. It occurs to powers 2 in H_1 and 4 in $\widehat{H}_2$ and $\widehat{H}_3$, consistent with the fact that the power of h_{10} in the denominator of j_2 and j_3 is 2 instead of 1 for j_1.

input degree	# multiplications	wall-clock time (s)
2	22 010	560
4	11 005	440
8	5 500	470
16	2 750	530
32	1 375	510
64	690	630
128	345	830
256	170	1 700
512	85	2 200
1 024	45	7 800
2 048	20	8 600
4 096	10	9 000
8 192	5	37 000
16 384	5	14 000

Table 8.4: Polynomial reconstruction timings for example with $\#\mathfrak{C} = 17\,608$.

Bibliography

[ABL+12] J. Anderson, J. S. Balakrishnan, K. Lauter, J. Park, and B. Viray. "Comparing arithmetic intersection formulas for denominators of Igusa class polynomials." 2012 (cit. on p. 12).

[Atk88] A. Atkin. "The number of points on an elliptic curve modulo a prime." In: *manuscript, Chicago IL* (1988) (cit. on p. 6).

[Bac90] E. Bach. "Explicit bounds for primality testing and related problems." In: *Math. Comp.* 55.191 (1990), pp. 355–380. ISSN: 0025-5718. DOI: 10.2307/2008811 (cit. on p. 47).

[Bel12] K. Belabas et al. PARI/GP. 2.5.3. http://pari.math.u-bordeaux.fr/. Bordeaux, Oct. 2012 (cit. on p. 57).

[BBE+08] J. Belding, R. Bröker, A. Enge, and K. Lauter. "Computing Hilbert Class Polynomials." In: *Algorithmic Number Theory, 8th International Symposium, ANTS-VIII, Banff, Canada, May 17-22, 2008, Proceedings*. Ed. by A. J. van der Poorten and A. Stein. Vol. 5011. Lecture Notes in Comput. Sci. Springer–Verlag, 2008, pp. 282–295 (cit. on pp. 8, 9, 56).

[BS09] G. Bisson and A. V. Sutherland. "Computing the endomorphism ring of an ordinary elliptic curve over a finite field." In: *Journal of Number Theory* (2009) (cit. on p. 49).

[Bis11] G. Bisson. "Endomorphism Rings in Cryptography." PhD thesis. Technische Universiteit Eindhoven and Institut National Polytechnique de Lorraine, July 2011. ISBN: 978-90-386-2519-5. Url: http://repository.tue.nl/714676 (cit. on p. 49).

[BCR12] G. Bisson, R. Cosset, and D. Robert. *AVIsogenies, a library for computing isogenies between abelian varieties*. http://avisogenies.gforge.inria.fr. 2012 (cit. on pp. 38, 41, 47).

[BS13] G. Bisson and M. Streng. "On polarised class groups of orders in quartic CM-fields." Preprint. 2013 (cit. on p. 49).

[Bor76] C.-W. Borchardt. "Das arithmetisch-geometrische Mittel aus vier Elementen." In: *Monatsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin* (Nov. 1876), pp. 611–621 (cit. on p. 29).

[Bor78] C.-W. Borchardt. "Theorie des arithmetisch-geometrischen Mittels aus vier Elementen." In: *Mathematische Abhandlungen der Königlichen Akademie der Wissenschaften zu Berlin* (1878), pp. 33–96 (cit. on p. 29).

[BB87] J. M. Borwein and P. B. Borwein. *Pi and the AGM*. John Wiley and Sons, 1987 (cit. on p. 31).

[Bre76] R. P. Brent. "Fast Multiple-Precision Evaluation of Elementary Functions." In: *Journal of the ACM* 23.2 (1976), pp. 242–251 (cit. on p. 28).

[BZ10] R. Brent and P. Zimmermann. *Modern Computer Arithmetic*. Vol. 18. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2010, 221 pages (cit. on p. 32).

[Brö08] R. Bröker. “A p-adic algorithm to compute the Hilbert class polynomial.” In: *Mathematics of Computation* 77.264 (2008), pp. 2417–2435 (cit. on p. 8).

[BGL11] R. Bröker, D. Gruenewald, and K. Lauter. “Explicit CM theory for level 2-structures on abelian surfaces.” In: *Algebra Number Theory* 5.4 (2011), pp. 495–528. DOI: 10.2140/ant.2011.5.495 (cit. on pp. 11, 20, 21, 42, 48, 56).

[BY06] J. H. Bruinier and T. Yang. “CM-values of Hilbert modular functions.” In: *Invent. Math.* 163.2 (2006), pp. 229–288. ISSN: 0020-9910. DOI: 10.1007/s00222-005-0459-7 (cit. on p. 12).

[CPS13] D. Cadé, X. Pujol, and D. Stehlé. FPLLL. 4.0.2. http://perso.ens-lyon.fr/damien.stehle/fplll/. Jan. 2013 (cit. on p. 57).

[CQ05] G. Cardona and J. Quer. “Field of moduli and field of definition for curves of genus 2.” In: ed. by T. Shaska. Vol. 13. Lecture Notes Series on Computing. Papers from the conference held at the University of Idaho, Moscow, ID, May 26–28, 2005. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005, pp. 71–83. ISBN: 981-256-459-4. DOI: 10.1142/9789812701640_0006 (cit. on pp. 7, 11).

[CKL08] R. Carls, D. Kohel, and D. Lubicz. “Higher-dimensional 3-adic CM construction.” In: *J. Algebra* 319.3 (2008), pp. 971–1006. ISSN: 0021-8693. DOI: 10.1016/j.jalgebra.2007.11.016 (cit. on pp. 11, 52).

[CL09] R. Carls and D. Lubicz. “A p-adic quasi-quadratic time point counting algorithm.” In: *Int. Math. Res. Not. IMRN* 4 (2009), pp. 698–735. ISSN: 1073-7928. DOI: 10.1093/imrn/rnn143 (cit. on pp. 11, 52).

[CHR08] J. Chaumine, J. Hirschfeld, and R. Rolland, eds. *Algebraic geometry and its applications*. Vol. 5. Series on Number Theory and its Applications. Dedicated to Gilles Lachaud on his 60th birthday. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2008, pp. xvi+513. ISBN: 978-981-279-342-3; 981-279-342-9.

[Coh93] H. Cohen. *A course in algorithmic algebraic number theory*. Vol. 138. Grad. Texts in Math. Springer–Verlag, 1993 (cit. on p. 21).

[CLP+11] A.-C. Cojocaru, K. Lauter, R. Pries, and R. Scheidler, eds. *WIN—women in numbers*. Vol. 60. Fields Institute Communications. Research directions in number theory, Including the proceedings of the Banff International Research Station (BIRS) Workshop held in Banff, AB, November 2–7, 2008. Providence, RI: American Mathematical Society, 2011, pp. xii+288. ISBN: 978-0-8218-5226-2.

[Cos11] R. Cosset. “Applications des fonctions thêta à la cryptographie sur courbes hyperelliptiques.” http://tel.archives-ouvertes.fr/tel-00642951. Thèse. Université Henri Poincaré - Nancy I, 2011 (cit. on pp. 30, 39).

[CR11] R. Cosset and D. Robert. *Computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of genus 2 curves*. Cryptology ePrint Archive, Report 2011/143. 2011. Url: http://eprint.iacr.org/2011/143 (cit. on p. 39).

[Deu58] M. Deuring. *Die Klassenkörper der komplexen Multiplikation*. Vol. 2. Teubner Stuttgart, 1958 (cit. on p. 6).

[Dic30] K. Dickman. “On the frequency of numbers containing prime factors of a certain relative magnitude.” In: *Ark. Mat. Astr. Fys.* 22A.10 (1930), pp. 1–14 (cit. on p. 56).

[Dup06] R. Dupont. “Moyenne arithmético-géométrique, suites de Borchardt et applications.” http://www.lix.polytechnique.fr/Labo/Regis.Dupont/these_soutenance.pdf. Thèse. École Polytechnique, 2006 (cit. on pp. 28–31, 58).

[Dup11] R. Dupont. “Fast evaluation of modular functions using Newton iterations and the AGM.” In: *Mathematics of Computation* 80.275 (2011), pp. 1823–1847 (cit. on pp. 28, 31).

[EL10] K. Eisenträger and K. Lauter. “A CRT algorithm for constructing genus 2 curves over finite fields.” In: ed. by F. Rodier and S. Vladut. Vol. 21. Séminaires et Congrès [Seminars and Congresses]. preprint version at arXiv:math/0405305 [math.NT]. Paris: Société Mathématique de France, 2010, pp. 161–176. ISBN: 978-2-85629-279-2 (cit. on pp. 11, 39, 40, 56).

[Elk97] N. Elkies. “Elliptic and modular curves over finite fields and related computational issues.” In: *Computational perspectives on number theory: proceedings of a conference in honor of AOL Atkin, September 1995, University of Illinois at Chicago*. Vol. 7. Amer Mathematical Society. 1997, p. 21 (cit. on p. 6).

[Eng09] A. Enge. “The complexity of class polynomial computation via floating point approximations.” In: *Mathematics of Computation* 78.266 (2009), pp. 1089–1107 (cit. on p. 8).

[ES10a] A. Enge and A. Sutherland. “Class invariants by the CRT method, ANTS IX: Proceedings of the Algorithmic Number Theory 9th International Symposium.” In: *Lecture Notes in Computer Science* 6197 (July 2010), pp. 142–156 (cit. on p. 10).

[Eng12] A. Enge. MPFRCX — *A library for univariate polynomials over arbitrary precision real or complex numbers*. 0.4.1. http://mpfrcx.multiprecision.org/. INRIA. July 2012 (cit. on p. 57).

[EGT+12] A. Enge, M. Gastineau, P. Théveny, and P. Zimmermann. GNU MPC — *A library for multiprecision complex arithmetic with exact rounding*. 1.0.1. http://mpc.multiprecision.org/. INRIA. Sept. 2012 (cit. on p. 57).

[EM03] A. Enge and F. Morain. “Fast Decomposition of Polynomials with Known Galois Group.” In: *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes — AAECC-15*. Ed. by M. Fossorier, T. Høholdt, and A. Poli. Vol. 2643. Lecture Notes in Computer Science. Berlin: Springer-Verlag, 2003, 254–264 (cit. on p. 57).

[ES10b] A. Enge and A. V. Sutherland. “Class invariants by the CRT method.” In: *Proceedings of the 9th Biennial International Symposium (ANTS-IX) held in Nancy, July 19–23, 2010*. Ed. by G. Hanrot, F. Morain, and E. Thomé. Vol. 6197. Lecture Notes in Computer Science. Berlin: Springer, 2010, pp. 142–156. ISBN: 978-3-642-14517-9; 3-642-14517-5. DOI: 10.1007/978-3-642-14518-6_14 (cit. on p. 53).

[ET13] A. Enge and E. Thomé. “Computing class polynomials for abelian surfaces.” In preparation. 2013 (cit. on pp. 2, 11).

[FLR11] J.-C. Faugère, D. Lubicz, and D. Robert. “Computing modular correspondences for abelian varieties.” In: *Journal of Algebra* 343.1 (Oct. 2011), pp. 248–277. DOI: 10.1016/j.jalgebra.2011.06.031. arXiv: 0910.4668 [cs.SC]. Url: http://www.normalesup.org/~robert/pro/publications/articles/modular.pdf. HAL: hal-00426338 (cit. on p. 52).

[FM02] M. Fouquet and F. Morain. "Isogeny volcanoes and the SEA algorithm." In: *Algorithmic number theory (Sydney, 2002)*. Vol. 2369. Lecture Notes in Comput. Sci. Berlin: Springer, 2002, pp. 276–291. DOI: 10.1007/3-540-45455-1_23 (cit. on p. 9).

[FL08] D. Freeman and K. Lauter. "Computing endomorphism rings of Jacobians of genus 2 curves over finite fields." In: *Proceedings of the 1st Symposium (SAGA) held in Papeete, May 7–11, 2007*. Ed. by J. Chaumine, J. Hirschfeld, and R. Rolland. Vol. 5. Series on Number Theory and its Applications. Dedicated to Gilles Lachaud on his 60th birthday. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2008, pp. 29–66. ISBN: 978-981-279-342-3; 981-279-342-9. DOI: 10.1142/9789812793430_0002 (cit. on pp. 11, 13, 39–43, 53, 56).

[Frö77] A. Fröhlich, ed. *Algebraic number fields: L-functions and Galois properties*. London: Academic Press [Harcourt Brace Jovanovich Publishers], 1977, pp. xii+704.

[GG99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge, England: Cambridge University Press, 1999 (cit. on pp. 32, 57).

[Gau04] P. Gaudry. "Algorithmes de comptage de points d'une courbe définie sur un corps fini." Preprint. 2004. Url: http://www.loria.fr/~gaudry/publis/pano.pdf (cit. on p. 8).

[GHK+06] P. Gaudry, T. Houtmann, D. Kohel, C. Ritzenthaler, and A. Weng. "The 2-adic CM method for genus 2 curves with application to cryptography." In: *Proceedings of the 12th International Conference on the Theory and Application of Cryptology and Information Security held in Shanghai, December 3–7, 2006*. Ed. by X. Lai and K. Chen. Vol. 4284. Lecture Notes in Computer Science. Berlin: Springer, 2006, pp. 114–129. ISBN: 978-3-540-49475-1; 3-540-49475-8. DOI: 10.1007/11935230_8 (cit. on pp. 11, 52).

[GS08] P. Gaudry and E. Schost. *Hyperelliptic curve point counting record: 254 bit Jacobian*. June 2008. Url: http://webloria.loria.fr/~gaudry/record127/ (cit. on p. 6).

[GTT+07] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. "A double large prime variation for small genus hyperelliptic index calculus." In: *Mathematics of Computation* 76.257 (2007), pp. 475–492 (cit. on p. 6).

[GKS11] P. Gaudry, D. R. Kohel, and B. A. Smith. "Counting Points on Genus 2 Curves with Real Multiplication." In: *ASIACRYPT*. Ed. by D. H. Lee and X. Wang. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 504–519. ISBN: 978-3-642-25384-3 (cit. on p. 6).

[GL12] E. Z. Goren and K. E. Lauter. "Genus 2 curves with complex multiplication." In: *Int. Math. Res. Not. IMRN* 5 (2012), pp. 1068–1142. ISSN: 1073-7928. DOI: 10.1093/imrn/rnr052 (cit. on pp. 12, 23–25).

[Got59] E. Gottschling. "Explizite Bestimmung der Randflächen des Fundamentalbereiches der Modulgruppe zweiten Grades." In: *Math. Ann.* 138 (1959), pp. 103–124 (cit. on p. 20).

[Gra13] T. Granlund et al. GMP — *The* GNU *Multiple Precision Arithmetic Library*. 5.1.1. http://gmplib.org/. Feb. 2013 (cit. on p. 57).

[Gru10] D. Gruenewald. "Computing Humbert surfaces and applications." In: *Arithmetic, Geometry, Cryptography and Codint Theory 2009* (2010), pp. 59–69 (cit. on p. 50).

[GJL+11] H. Grundman, J. Johnson-Leung, K. Lauter, A. Salerno, B. Viray, and E. Wittenborn. "Igusa class polynomials, embeddings of quartic CM fields, and arithmetic intersection theory." In: ed. by A.-C. Cojocaru, K. Lauter, R. Pries, and R. Scheidler. Vol. 60. Fields Institute Communications. Research directions in number theory, Including the proceedings of the Banff International Research Station (BIRS) Workshop held in Banff, AB, November 2–7, 2008. Providence, RI: American Mathematical Society, 2011, pp. 35–60. ISBN: 978-0-8218-5226-2 (cit. on p. 12).

[HLP+12] G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann et al. GNU MPFR — *A library for multiple-precision floating-point computations with exact rounding*. 3.1.1. `http://www.mpfr.org/`. July 2012 (cit. on p. 57).

[HMT10] G. Hanrot, F. Morain, and E. Thomé, eds. *Algorithmic number theory*. Vol. 6197. Lecture Notes in Computer Science. Berlin: Springer, 2010, front matter+397. ISBN: 978-3-642-14517-9; 3-642-14517-5. DOI: `10.1007/978-3-642-14518-6`.

[Igu62] J.-I. Igusa. "On Siegel Modular Forms of Genus Two." In: *American Journal of Mathematics* 84 (1962), pp. 175–200 (cit. on p. 16).

[Igu72] J.-I. Igusa. *Theta functions*. Vol. 194. Die Grundlehren der mathematischen Wissenschaften. Springer, 1972 (cit. on p. 30).

[IJ10] S. Ionica and A. Joux. "Pairing the volcano." In: *Algorithmic Number Theory* (2010), pp. 201–218 (cit. on pp. 10, 50).

[Ion12] S. Ionica. "Pairing-based algorithms for jacobians of genus 2 curves with maximal endomorphism ring." In: *IACR Cryptology ePrint Archive* 2012 (2012), p. 167 (cit. on p. 50).

[Kli90] H. Klingen. *Introductory lectures on Siegel modular forms*. Vol. 20. Cambridge studies in advanced mathematics. Cambridge University Press, 1990 (cit. on p. 32).

[Koh96] D. Kohel. "Endomorphism rings of elliptic curves over finite fields." PhD thesis. University of California, 1996 (cit. on p. 9).

[LO77] J. C. Lagarias and A. M. Odlyzko. "Effective versions of the Chebotarev density theorem." In: *Proceedings of a Symposium held at the University of Durham, Durham, Sept. 2–12, 1975*. Ed. by A. Fröhlich. London: Academic Press [Harcourt Brace Jovanovich Publishers], 1977, pp. 409–464 (cit. on pp. 9, 56).

[LC06] X. Lai and K. Chen, eds. *Advances in cryptology—ASIACRYPT 2006*. Vol. 4284. Lecture Notes in Computer Science. Berlin: Springer, 2006, pp. xiv+468. ISBN: 978-3-540-49475-1; 3-540-49475-8. DOI: `10.1007/11935230`.

[LR12] K. Lauter and D. Robert. "Improved CRT Algorithm for class polynomials in genus 2." In: *ANTS* (2012). Accepted for publication at the Tenth Algorithmic Number Theory Symposium ANTS-X. University of California, San Diego, July 9 – 13, 2012 `http://math.ucsd.edu/~kedlaya/ants10/`. Longer version available on eprint. Slides `http://www.normalesup.org/~robert/publications/slides/2012-07-ANTS-SanDiego.pdf`, eprint: 2012/443, HAL: hal-00734450 (cit. on pp. 2, 11, 39, 41, 55, 56).

[LV12] K. Lauter and B. Viray. "An arithmetic intersection formula for denominators of Igusa class polynomials." Preprint ArXiV 1210.7841. 2012 (cit. on p. 12).

[LY11] K. Lauter and T. Yang. "Computing genus 2 curves from invariants on the Hilbert moduli space." In: *J. Number Theory* 131.5 (2011), pp. 936–958. ISSN: 0022-314X. DOI: 10.1016/j.jnt.2010.05.012 (cit. on pp. 37, 50).

[LPP02] H. W. Lenstra Jr., J. Pila, and C. Pomerance. "A hyperelliptic smoothness test. II." In: *Proc. London Math. Soc. (3)* 84.1 (2002), pp. 105–146. ISSN: 0024-6115. DOI: 10.1112/plms/84.1.105 (cit. on p. 54).

[LL03] R. Lercier and D. Lubicz. "Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time." In: *Advances in Cryptology—EUROCRYPT '2003*. Ed. by E. Biham. Lecture Notes in Computer Science. Springer-Verlag, May 2003 (cit. on p. 8).

[LR10] D. Lubicz and D. Robert. *Computing isogenies between abelian varieties*. to appear in *Compositio Mathematica*. 2010 (cit. on pp. 39, 52).

[Mes91] J.-F. Mestre. "Construction de courbes de genre 2 à partir de leurs modules." In: *Effective methods in algebraic geometry*. Ed. by T. Mora and C. Traverso. Vol. 94. Progr. Math. Birkhäuser, 1991, 313–334 (cit. on pp. 7, 11, 57).

[Mes02] J.-F. Mestre. *Notes of a talk given at the Cryptography Seminar Rennes*. 2002. Url: http://www.math.univ-rennes1.fr/crypto/2001-02/mestre.ps (cit. on p. 52).

[Mil06] J. S. Milne. *Complex multiplication*. Online notes available at http://www.jmilne.org/math/CourseNotes/cm.html. 2006 (cit. on pp. 11, 23).

[MT91] T. Mora and C. Traverso, eds. *Effective methods in algebraic geometry*. Vol. 94. Progress in Mathematics. Papers from the symposium (MEGA-90) held in Castiglioncello, April 17–21, 1990. Boston, MA: Birkhäuser Boston Inc., 1991, pp. xiv+500. ISBN: 0-8176-3546-7. DOI: 10.1007/978-1-4612-0441-1.

[Pil90] J. Pila. "Frobenius maps of abelian varieties and finding roots of unity in finite fields." In: *Mathematics of Computation* 55.192 (1990), pp. 745–763 (cit. on p. 6).

[PS08] A. J. van der Poorten and A. Stein, eds. *Algorithmic number theory*. Vol. 5011. Lecture Notes in Computer Science. Berlin: Springer, 2008, pp. x+455. ISBN: 978-3-540-79455-4; 3-540-79455-7. DOI: 10.1007/978-3-540-79456-1.

[Rob10] D. Robert. "Fonctions thêta et applications à la cryptographie." PhD thesis. Université Henri Poincaré — Nancy 1, July 2010. Url: http://hal.inria.fr/tel-00528942/ (cit. on p. 39).

[Rob13] D. Robert. "Computing cyclic isogenies using real multiplication." ANR Peace Meeting, Paris. Notes available on http://www.normalesup.org/~robert/pro/publications/notes/2013-04-cyclic-isogenies.pdf. Slides. Apr. 2013 (cit. on p. 50).

[RV10] F. Rodier and S. Vladut, eds. *Arithmetics, geometry, and coding theory (AGCT 2005)*. Vol. 21. Séminaires et Congrès [Seminars and Congresses]. Papers from the conference held in Marseilles, September 26–30, 2005. Paris: Société Mathématique de France, 2010, pp. xxii+225. ISBN: 978-2-85629-279-2.

[Sat00] T. Satoh. "The canonical lift of an ordinary elliptic curve over a finite field and its point counting." In: *J. Ramanujan Math. Soc.* 15.4 (2000), pp. 247–270 (cit. on p. 8).

[Sch02] R. Schertz. "Weber's class invariants revisited." In: *Journal de théorie des nombres de Bordeaux* 14.1 (2002), pp. 325–343 (cit. on p. 13).

[Sch85] R. Schoof. "Elliptic curves over finite fields and the computation of square roots mod p." In: *Mathematics of computation* 44.170 (1985), pp. 483–494 (cit. on p. 6).

[Sha05] T. Shaska, ed. *Computational aspects of algebraic curves*. Vol. 13. Lecture Notes Series on Computing. Papers from the conference held at the University of Idaho, Moscow, ID, May 26–28, 2005. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005, pp. xii+272. ISBN: 981-256-459-4.

[Shi98] G. Shimura. *Abelian varieties with complex multiplication and modular functions*. Vol. 46. Princeton Mathematical Series. Princeton, NJ: Princeton University Press, 1998, pp. xvi+218. ISBN: 0-691-01656-9 (cit. on pp. 6, 14, 23).

[ST61] G. Shimura and Y. Taniyama. *Complex Multiplication of Abelian Varieties and its Applications to Number Theory*. The Mathematical Society of Japan, 1961 (cit. on p. 14).

[Spa94] A.-M. Spallek. "Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen." PhD thesis. Universität Gesamthochschule Essen, 1994 (cit. on pp. 11, 12, 16).

[Str] M. Streng. "An explicit version of Shimura's reciprocity law for Siegel modular functions." Preprint (cit. on p. 13).

[Str09] M. Streng. "Computing Igusa class polynomials." To appear in *Mathematics of Computation*. 2009. Url: http://arxiv.org/abs/0903.4766 (cit. on pp. 14, 59).

[Str10] M. Streng. "Complex multiplication of abelian surfaces." Proefschrift. Universiteit Leiden, 2010 (cit. on pp. 11, 12, 14–16, 18, 23, 53, 59).

[Sut] A. Sutherland. *Genus 1 point counting in quadratic space and essentially quartic time*. Url: www-math.mit.edu/~drew/CRMPointCounting0410.pdf (cit. on p. 6).

[Sut10] A. Sutherland. "Computing Hilbert class polynomials with the Chinese remainder theorem." In: *Mathematics of Computation* 80.273 (2010), pp. 501–538 (cit. on p. 9).

[Vél71] J. Vélu. "Isogénies entre courbes elliptiques." In: *Compte Rendu Académie Sciences Paris Série A-B* 273 (1971), A238–A241 (cit. on p. 8).

[Wam99] P. van Wamelen. "Examples of genus two CM curves defined over the rationals." In: *Math. Comp.* 68.225 (1999), pp. 307–320. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-99-01020-0 (cit. on p. 11).

[Wen03] A. Weng. "Constructing hyperelliptic curves of genus 2 suitable for cryptography." In: *Math. Comp.* 72.241 (2003), 435–458 (electronic). ISSN: 0025-5718. DOI: 10.1090/S0025-5718-02-01422-9 (cit. on p. 11).

[Yan10a] T. Yang. "An arithmetic intersection formula on Hilbert modular surfaces." In: *Amer. J. Math.* 132.5 (2010), pp. 1275–1309. ISSN: 0002-9327. DOI: 10.1353/ajm.2010.0002 (cit. on p. 12).

[Yan10b] T. Yang. *Arithmetic Intersection on a Hilbert Modular Surface and the Faltings Height*. 2010 (cit. on p. 12).