Computing class polynomials for abelian surfaces

Andreas Enge* and Emmanuel Thomé†

9 December 2013

Abstract

We describe a quasi-linear algorithm for computing Igusa class polynomials of Jacobians of genus 2 curves via complex floating-point approximations of their roots. After providing an explicit treatment of the computations in quartic CM fields and their Galois closures, we pursue an approach due to Dupont for evaluating ϑ-constants in quasi-linear time using Newton iterations on the Borchardt mean. We report on experiments with our implementation and present an example with class number 20016.

1 Introduction

Igusa class polynomials describe the complex multiplication points in the moduli space of principally polarised abelian surfaces, that is, they parameterise abelian varieties of dimension 2 with complex multiplication by a maximal order of a quartic CM field. Such abelian surfaces are Jacobians of hyperelliptic curves of genus 2, so that by computing Igusa class polynomials one may obtain genus 2 curves over finite fields with known Jacobian cardinality.

In the dimension 1 case of elliptic curves, several approaches have been described in the literature. While the output of the algorithms (a large polynomial) is of exponential size in the input (a number field described by a single integer), all of these approaches may lead to an algorithm with a complexity that is quasi-linear (up to logarithmic factors) in its output size: The complex analytic method uses floating point approximations to the roots of the class polynomials [16]; the p-adic approach starts from a curve with the given endomorphism ring over a small finite field and lifts its invariants to a p-adic field [13, 8]; the Chinese remaindering approach combines curves over several small prime fields [2].

*INRIA, LFANT, F-33400 Talence, France
CNRS, IMB, UMR 5251, F-33400 Talence, France
Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France
andreas.enge@inria.fr

†INRIA, CARAMEL, Nancy, France
emmanuel.thome@inria.fr

In principle, the same approaches apply to abelian surfaces. A 2-adic algorithm is described in [21], and there are currently attempts at making the Chinese remainder based method more efficient [28]. So far, 2-adic lifting appears to have been the most successful approach: The Echidna database maintained by Kohel[1] contains Igusa class polynomials, the largest of which is of degree 576 and has been obtained by lifting from a curve over $\mathbb{F}_{2^6}$[2].

A detailed description of the complex analytic approach, together with complexity analyses of its different steps, has recently been given in [33, 34]. Our work pushes the limits for the attainable degrees of Igusa class polynomials: We present an example of degree 20 016. Moreover, relatively small class polynomials (say, below degree 150) can be computed in matters of seconds. The key tool in this approach is the use of a quasi-linear algorithm for the computation of ϑ-constants, initially described in [15].

This article is organised as follows. §2 presents the necessary background material for discussing the complex multiplication theory of abelian surfaces and states the general algorithm. §§3 and 4 show how to explicitly (providing concrete descriptions for the occurring number fields, maps between them and their embeddings) and symbolically compute an appropriate set of reduced period matrices, which form the input of the computationally expensive step of computing ϑ-constants, detailed in §5. The recognition as algebraic numbers of the coefficients of the Igusa class polynomials from their approximations by complex embeddings is described in §6, and experimental results are given in §§7 and 8.

All computations presented in this article have been achieved with the software package Cmh[20], released under the GNU General Public License.

2 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 our algorithms and implementation. The presentation follows [34], and proofs are given in [31, 30, 34, 33].

2.1 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 κ 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 $\kappa \circ \varphi = \varphi \circ \kappa$, which justifies the notation $\overline{\varphi} = \kappa \circ \varphi$.

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. The Galois closure L is itself a CM field, and the complex conjugation of L, which we denote again by κ, restricts to the complex conjugation

[1] `http://echidna.maths.usyd.edu.au/echidna/dbs/complex_multiplication2.html`

[2] Personal communication

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

$$\mathrm{N}_{\Phi^r}\,\overline{\mathrm{N}}_{\Phi^r} = \mathrm{N}; \tag{1}$$

this map extends to ideals and ideal classes.

In §3, 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 $\delta_K^{-1} = \{y \in K : \mathrm{Tr}(xy) \in \mathbb{Z}\ \forall x \in \mathcal{O}_K\}$ be the codifferent ideal of K. Assume that $(\mathfrak{a}\overline{\mathfrak{a}}\delta_K)^{-1}$ 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 $A(\Phi, \mathfrak{a}, \xi) = \big(\mathbb{C}^2/\Phi(\mathfrak{a}), E_{\Phi,\xi}\big)$ 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 $\mathcal{O}_K$.

2.2 The Shimura group, its type norm subgroup and cosets

The Igusa invariants to be defined in §2.3 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. 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 $\mathrm{Gal}(K_0^r/\mathbb{Q})$ [34, Lemmata 1.1 and 2.1].

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

$$\mathfrak{C} = \big\{(\mathfrak{a}, u) : \mathfrak{a} \text{ a fractional ideal of } \mathcal{O}_K, \mathfrak{a}\overline{\mathfrak{a}} = u\mathcal{O}_K, \text{ and } u \in K_0 \text{ totally positive}\big\}/\sim \tag{2}$$

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, the Shimura class group $\mathfrak{C}$ acts regularly on $\mathcal{M}_{K,\Phi}$ via

$$(\mathfrak{b}, u) \cdot A(\Phi, \mathfrak{a}, \xi) = A(\Phi, \mathfrak{b}^{-1}\mathfrak{a}, u\xi). \tag{3}$$

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 (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 [34, 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 [34, Theorem 2.2], so $\mathcal{M}_{K,\Phi}$ splits into orbits under $\mathfrak{C}$ of size $|\operatorname{im}(\mathrm{N}_{\Phi^r})|$, 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 $\mathrm{Gal}(K_0^r/\mathbb{Q})$.

2.3 ϑ-functions, Igusa invariants and class polynomials

Given an ideal $\mathfrak{a}$ and a principal polarisation $E_{\Phi,\xi}$ as in §2.1, 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*

$$\Omega_{\Phi,\mathfrak{a},\xi} = W^{-1}V \tag{4}$$

in the *Siegel half space* $\mathcal{H}_2 = \{\Omega \in \mathbb{C}^{2\times 2} : \Omega \text{ symmetric and } \Im(\Omega) \text{ positive definite}\}$. The symplectic group $\mathrm{Sp}_4(\mathbb{Z})$ acts on $\mathcal{H}_2$ 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 $\mathcal{H}_2$ exists under the action of $\mathrm{Sp}_4(\mathbb{Z})$. Reduction into the fundamental domain is discussed in §3.3.

The 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 $\Omega \in \mathcal{H}_2$,

$$\vartheta_{16a_1+8a_2+4b_1+2b_2}(\Omega) = \vartheta_{a,b}(\Omega) = \sum_{n \in \mathbb{Z}^2} e^{2\pi i \left(\frac{1}{2}(n+a)^\intercal \Omega (n+a) + (n+a)^\intercal b\right)}. \tag{5}$$

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).

$$\begin{aligned}
4\vartheta_0^2(\Omega) &= \vartheta_0^2 + \vartheta_1^2 + \vartheta_2^2 + \vartheta_3^2 & 4\vartheta_6^2(\Omega) &= 2\vartheta_0\vartheta_2 - 2\vartheta_1\vartheta_3 \\
4\vartheta_1^2(\Omega) &= 2\vartheta_0\vartheta_1 + 2\vartheta_2\vartheta_3 & 4\vartheta_8^2(\Omega) &= \vartheta_0^2 + \vartheta_1^2 - \vartheta_2^2 - \vartheta_3^2 \\
4\vartheta_2^2(\Omega) &= 2\vartheta_0\vartheta_2 + 2\vartheta_1\vartheta_3 & 4\vartheta_9^2(\Omega) &= 2\vartheta_0\vartheta_1 - 2\vartheta_2\vartheta_3 \\
4\vartheta_3^2(\Omega) &= 2\vartheta_0\vartheta_3 + 2\vartheta_1\vartheta_2 & 4\vartheta_{12}^2(\Omega) &= \vartheta_0^2 - \vartheta_1^2 - \vartheta_2^2 + \vartheta_3^2 \\
4\vartheta_4^2(\Omega) &= \vartheta_0^2 - \vartheta_1^2 + \vartheta_2^2 - \vartheta_3^2 & 4\vartheta_{15}^2(\Omega) &= 2\vartheta_0\vartheta_3 - 2\vartheta_1\vartheta_2
\end{aligned} \tag{6}$$

Denote by h_j the following modular forms of weight j:

$$\begin{aligned}
h_4 &= \sum_{i \in T} \vartheta_i^8, \quad h_6 = \sum_{60 \text{ triples } (i,j,k) \in T^3} \pm(\vartheta_i\vartheta_j\vartheta_k)^4, \\
h_{10} &= \prod_{i \in T} \vartheta_i^2, \quad h_{12} = \sum_{15 \text{ tuples } (i,j,k,l,m,n) \in T^6} (\vartheta_i\vartheta_j\vartheta_k\vartheta_l\vartheta_m\vartheta_n)^4;
\end{aligned} \tag{7}$$

for the exact definitions, see [34, §II.7.1]. These generate the ring of holomorphic Siegel modular forms over $\mathbb{C}$, see [25, Corollary p. 195] and [34, 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 $\frac{h_{12}^5}{h_{10}^6}, \frac{h_{12}^3 h_4}{h_{10}^4}, \frac{h_{12}^2 h_6}{h_{10}^3}$ [32, Satz 5.2]. Streng defines invariants with the minimal powers of h_{10} in the denominator as

$$j_1 = \frac{h_4 h_6}{h_{10}}, \quad j_2 = \frac{h_4^2 h_{12}}{h_{10}^2}, \quad j_3 = \frac{h_4^5}{h_{10}^2}. \tag{8}$$

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 §2.2. Following the discussion there, the *Igusa class polynomials* $I_i(X) =$

$\prod_{(\Phi,\mathfrak{a},\xi)}(X - j_i(\Omega_{\Phi,\mathfrak{a},\xi}))$ 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

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

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 §3) and a base point $\beta = (\Phi, \mathfrak{a}_\Phi, \xi_\Phi)$ and let

$$H_1(X) = \prod_{C \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})} (X - j_1(\Omega_{C\cdot\beta})). \tag{9}$$

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 $\hat{H}_2$ and $\hat{H}_3$ through $j_i H_1'(j_1) = \hat{H}_i(j_1)$ with

$$\hat{H}_i(X) = \sum_{C \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r})} j_i(\Omega_{C\cdot\beta}) \prod_{D \in \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) \setminus \{C\}} (X - j_1(\Omega_{D\cdot\beta})) \tag{10}$$

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

2.4 Algorithm for Igusa class polynomials

We briefly summarise the algorithm for computing class polynomials.

Algorithm 1

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

OUTPUT: *Irreducible class polynomials H_1, $\hat{H}_2$, $\hat{H}_3 \in K_0^r[X]$ 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 (6), the values* $h_k(\Omega_i)$ *by (7) and finally the triples* $J_i = (j_1(\Omega_i), j_2(\Omega_i), j_3(\Omega_i))$ *by (8).*
5) *Let* $H_1 = \prod_{i=1}^h (X - J_{i,1})$, $\hat{H}_k = \sum_{i=1}^h J_{i,k} \prod_{l \neq i}(X - J_{l,1}) \in \mathbb{C}[X]$ *for* $k \in \{2, 3\}$.
6) *Recognise the coefficients of* H_1, $\hat{H}_2$, $\hat{H}_3$ *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) are described in §3. Step 1) is treated in §4.2, Step 4) in §5 and Step 6) in §6.

3 Explicit equations and symbolic period matrices

While Algorithm 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 §3.1 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 §3.2 and derive explicit equations for all considered fields and embeddings.

Recall the notation of §2: 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.

3.1 Galois theory, embeddings and period matrices

3.1.1 The dihedral case

Galois theory. Let $K = \mathbb{Q}(y)$ be a non-Galois quartic CM field. The following statements are easily seen to be true when choosing a generator y such that $z = y^2$ belongs to the real subfield K_0, so that $K = \mathbb{Q}[Y]/(Y^4 + AY^2 + B)$ for some $A, B \in \mathbb{Q}$. The Galois closure of K is then $L = K(y') = \mathbb{Q}(y, y')$, where the roots of the minimal polynomial of y in L are $\pm y$ and $\pm y'$ (the former could be identified with $\varphi_1(y)$ in (12), the latter with $\varphi_2(y)$). The automorphisms in $G = \mathrm{Gal}(L/\mathbb{Q})$ are uniquely determined by their images on y and y', and we obtain the following diagram of fields and Galois groups:

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

Here the automorphisms are given by

$$\begin{aligned}
&\rho : y \mapsto y,\ y' \mapsto -y' \text{ of order } 2\\
&\sigma : y \mapsto y',\ y' \mapsto y \text{ of order 2, which fixes the generator } y+y' \text{ of } K^r\\
&\tau : y \mapsto y',\ y' \mapsto -y \text{ of order } 4\\
&\kappa = \tau^2 : y \mapsto -y,\ y' \mapsto -y \text{ is the complex conjugation.}
\end{aligned}$$

So G is the dihedral group D_4 with generators τ of order 4 and ρ (or σ) of order 2 and additional relation $\rho\tau\rho = \tau^3$, and with $\langle\kappa\rangle$ as its centre.

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, $\kappa = \tau^2$ nor $\kappa\rho = \tau^2\rho$. 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 §2.3. 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

$$\Omega_{\Phi,\mathfrak{a},\xi} = W^{-1}V = \psi(M) \text{ with } M = \frac{1}{\alpha_3\alpha_4^\sigma - \alpha_4\alpha_3^\sigma}\begin{pmatrix} \alpha_4\alpha_1^\sigma - \alpha_1\alpha_4^\sigma & \alpha_4\alpha_2^\sigma - \alpha_2\alpha_4^\sigma \\ \alpha_3\alpha_1^\sigma - \alpha_1\alpha_3^\sigma & \alpha_3\alpha_2^\sigma - \alpha_3\alpha_2^\sigma \end{pmatrix} \tag{11}$$

by (4). The entries of M are invariant under σ and thus elements of K^r.

Remark. 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 $\mathrm{Gal}\,(L/(K^r)') = \langle\kappa\sigma\rangle = \langle\rho\sigma\rho\rangle$.

3.1.2 The cyclic case

Here we have the much simpler situation

$$\begin{array}{l} K \\ \big| \ \langle \kappa \rangle = \langle \sigma^2 \rangle \\ K_0 \\ \big| \\ \mathbb{Q} \end{array}$$

We may choose $\psi = \varphi_1$. Then there is a uniquely determined $\sigma \in \mathrm{Gal}(K/\mathbb{Q})$ such that $\varphi_2 = \varphi_1 \sigma$, and trivially M of (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.

3.2 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 (4). We use the same notation for elements of the Galois group G of $L/\mathbb{Q}$ as in §3.1.

3.2.1 The dihedral case

Field equations. By choosing generating elements as in §3.1.1 we may assume that

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

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

$$\varphi_1(y) = i\sqrt{\frac{A + \sqrt{A^2 - 4B}}{2}}, \quad \varphi_2(y) = i\sqrt{\frac{A - \sqrt{A^2 - 4B}}{2}}, \tag{12}$$

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 §3.1.1 the notations $\mathrm{Gal}(L/K) = \langle \rho \rangle$, $\mathrm{Gal}(L/K^r) = \langle \sigma \rangle$, 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

$$\psi(y^r) = \psi(y) + (\psi\sigma)(y) = \varphi_1(y) + \varphi_2(y) \tag{13}$$

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 over K or K^r 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 (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 us 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 [9, §3.1]. Computing dual type norms thus reduces to conversions in relative extensions as described above.

3.2.2 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 (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 [9, §3.1].

3.3 Symbolic reduction of period matrices

Gottschling in [22] 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 §2.3 are modular for $\mathrm{Sp}_4(\mathbb{Z})$, we may transform all period matrices occurring in Algorithm 1 into $\mathcal{F}_2$. A period matrix Ω is *reduced* if $\Re(\Omega)$ 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 $\Im(\Omega)$ 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 (11), $\Omega = \psi(M)$ with $M \in (K^r)^{2\times 2}$ and an explicitly given $\psi : K^r \to \mathbb{C}$, see (13) and (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 $\mathcal{F}_2$ 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 $b^2 D^r$.

4 Computing the Shimura group and its type norm subgroup

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

The first step of Algorithm 1 requires to enumerate the Shimura group $\mathfrak{C}$ of (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 [9]:

$$1 \longrightarrow \mathcal{O}_{K_0}^+ / N_{K/K_0}(\mathcal{O}_K^*) \xrightarrow{u \mapsto (\mathcal{O}_K, u)} \mathfrak{C} \xrightarrow{(\mathfrak{a}, \alpha) \mapsto \mathfrak{a}} \mathrm{Cl}_K \xrightarrow{N_{K/K_0}} \mathrm{Cl}_{K_0}^+ \longrightarrow 1, \qquad (14)$$

where $\mathcal{O}_{K_0}^+$ is the subgroup of totally positive units in $\mathcal{O}_{K_0}$ 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 $\mathcal{O}_{K_0}^+ / \mathrm{N}_{K/K_0}(\mathcal{O}_K^*)$. Let ε_0 and ε be the fundamental units of K_0 and K, respectively. If $\mathrm{N}(\varepsilon_0) = -1$, then $\mathcal{O}_{K_0}^+ = \langle \varepsilon_0^2 \rangle = \mathrm{N}_{K/K_0}(\langle \varepsilon_0 \rangle) \subseteq \mathrm{N}_{K/K_0}(\mathcal{O}_K^*)$, and the quotient group is trivial. If $\mathrm{N}(\varepsilon_0) = +1$, then $\mathcal{O}_{K_0}^{+} = \langle \varepsilon_0 \rangle$, and since $\varepsilon_0^2 = \mathrm{N}_{K/K_0}(\varepsilon_0) \in \mathrm{N}_{K/K_0}(\mathcal{O}_K^*)$, 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 (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 [11, §2.4].

Algorithm 2
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 the generators* $(\mathfrak{a}_1, \alpha_1), \ldots, (\mathfrak{a}_{r'}, \alpha_{r'})$ *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 (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.

4.2 The type norm subgroup

Algorithm 2 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 §2.3.

5 Computing ϑ-constants

As explained in Step 4) of Algorithm 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 $\Omega = \begin{pmatrix} \omega_0 & \omega_1 \\ \omega_2 & \omega_0 \end{pmatrix} \in \mathcal{F}_2$.

In §5.1 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 §§5.2–5.4 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 [15]; for the corresponding algorithm in dimension 1, using the arithmetic-geometric mean instead of the Borchardt mean, see [14]. We provide a streamlined presentation in dimension 2, together with improved algorithms and justifications.

5.1 Naive approach

For the fundamental ϑ-constants, (5) specialises as

$$\vartheta_{4b_1+2b_2}(\Omega/2) = \sum_{m,n\in\mathbb{Z}} (-1)^{2(mb_1+nb_2)} q_0^{m^2} q_1^{2mn} q_2^{n^2} \tag{15}$$

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 $R \to \infty$; [15, 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 multiplications of complex numbers.

Proposition 3 *The truncated sum over $(m, n) \in [-R, R]^2$ for the fundamental ϑ-constants* (15) *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

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

where $\tilde{O}(N) = O\left(N(\log N)^{O(1)}\right)$, *and* $\mathrm{M}(N) \in \tilde{O}(N)$ *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

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

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 [7]. $\square$

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

5.2 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 [3, 4]. The complex case is treated in [15], 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 *Let*

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

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

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

Definition and Properties 5 *Given a complex quadruple $b = (b_0, \dots, b_3) \in \mathbb{C}^4$, a* Borchardt iterate *is a quadruple $b' = (b'_0, \dots, b'_4)$ such that there are four choices of square roots $(\sqrt{b_j})_{j=0,\dots,3}$ 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 $b \in \mathcal{H}^4$, the standard Borchardt iterate *is obtained by choosing square roots in $\mathcal{Q}$, so that $b' \in \mathcal{H}^4$ again. More generally, if all entries of b lie in the same half-plane, that is, $b \in (z\mathcal{H})^4$ for some $z \in \mathbb{C}$, choosing all square roots in the same quarter-plane $\sqrt{z}\mathcal{Q}$ (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 [15, Chapter 7].

Proposition 6 *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 precision of N bits with $O(\log N)$ multiplications in time

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

Comparison of the formulæ in Definition 5 and (6) shows that for any period matrix $\Omega \in \mathcal{H}_2$, the sequence $\left((\vartheta_j^2(2^n\Omega))_{j=0,\dots,3}\right)_{n \geqslant 0}$ 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 $\Omega \in \mathcal{F}_2$.

Proposition 7 *For $\Omega \in \mathcal{F}_2$, $n \geqslant 0$ and $j = 0, \dots, 3$ we have $\vartheta_j(2^n\Omega) \in \mathcal{Q}$. Hence $\left((\vartheta_j^2(2^n\Omega))_{j=0,\dots,3}\right)_{n \geqslant 0}$ is the standard Borchardt sequence associated to $(\vartheta_j^2(\Omega))_{j=0,\dots,3}$. It converges to 1.*

The result follows from [15, Propositions 6.1 and 9.1].

5.3 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 [15, §9.2.3], which lead to well-behaved Borchardt means, see Conjecture 9.

Proposition 8 *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* $\Omega \in \mathcal{H}_2$. *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 [12, Propriété 3.1.24], following [26, Chapter 5, Theorem 2]. The explicit form restricted to squares of ϑ-constants, as given here, is found in [15, §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 9 *Let*

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

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

Under Conjecture 9, 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 10
Input: *Floating point approximations of* $(\vartheta_j(\Omega/2)/\vartheta_0(\Omega/2))_{j=1,2,3}$ *for some* $\Omega \in \mathcal{F}_2$*, and as auxiliary data the sign of* ω_1.
Output: *Floating point approximations of the coefficients* ω_0*,* ω_1*,* ω_2 *of* $\Omega \in \mathcal{F}_2$

1) *Use the duplication formulæ* (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 10 under Conjecture 9 follows from the discussions above. Step 1) uses the homogeneity of (6), Step 2) the homogeneity of the Borchardt mean and Proposition 7. The validity of Step 5) follows from Proposition 8 under Conjecture 9, 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 $\Omega \in \mathcal{F}_2$, the fundamental domain for all of $\mathrm{Sp}_4(\mathbb{Z})$, it is necessarily unique; Conjecture 9 implies that this particular representative for Ω is indeed returned by the algorithm.

5.4 Newton lift for fundamental ϑ-constants

Denote by

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

the function computed by Algorithm 10, and by

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

its inverse on $\mathcal{F}_2$ (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 10 may be modified accordingly to also output J_F, see [15, Algorithme 16], generalising the dimension 1 approach of [5, §2.4] and [14]. 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 §7.1).

Algorithm 11
INPUT: *Floating point approximations* $y^{(n)}$ *of* $\Omega \in \mathcal{F}_2$, *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 (see Theorem 12).*

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 10, 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 12 *Assume the validity of Conjecture 9. Let* $\Omega \in \mathcal{F}_2$ *be such that* $\vartheta_0(\Omega/2) \neq 0$, $x = f(\Omega)$, *and let* $x^{(0)}$ *be 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 11 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 [35, Chapter 9 and §15.4] and [6, §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 6. □

Notice that for our application of computing class polynomials for primitive quartic CM fields, the assumption of Theorem 12 is satisfied: As $\begin{pmatrix}\Omega & \mathrm{id}_2\end{pmatrix}\mathbb{Z}^4$ is of rank 4, we have $\omega_1 \neq 0$, and therefore none of the $\vartheta_j(\Omega/2)$ vanish (see [27, Chapter 9, Proposition 2]).

In practice, we use a fixed initial precision for $x^{(0)}$, computed according to Proposition 3, 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. It is possible to modify Algorithm 10 and consequently Algorithm 11 so as not to rely on Conjecture 9. 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 §5.1, one can make sure to choose the correct sign. These computations do not deteriorate the asymptotic complexity; moreover, as Algorithm 11 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 9 with tens of thousands of arguments.

6 Reconstruction of class polynomial coefficients and reduction modulo prime ideals

6.1 The dihedral case

The class polynomials H_1, $\hat{H}_2$, $\hat{H}_3$ of (9) and (10) for a fixed CM type Φ are defined over K_0^r, but Steps 1) to 5) of Algorithm 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 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. §3.2. We may limit the discussion to the CM type Φ and the embedding ψ given by (13) and (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 $2^{n+e^-}\left|c\beta - (a + b\sqrt{D^r})\right| \approx 2^{e^+}c$ 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 (13) and (12),

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

so that

$$w = \sqrt{\frac{D^r}{B}} \cdot \frac{-z^r - A}{2}. \tag{16}$$

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.

6.2 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.

7 Implementation and parallelisation

Our implementation of the algorithms defined here is available in the software package Cmh[20], which can be downloaded from

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

The current version of the Cmh software package is still in development, and will be named Cmh-1.0 once some packaging improvements, alongside with minor bug corrections, are checked in.

The software implements the different steps of Algorithm 1 as follows:

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

- The computation of ϑ-constants in Step 4) of Algorithm 1 is done by a C program, based on the library Gnu Mpc[18], itself using the Gnu Mpfr[24] and Gnu Mp[23] libraries. Newton lifting is used for this step from a base precision of 2 000 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[17] 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 $\hat{H}_k$, $k = 2, 3$, 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 $\hat{H}_k$, $k = 2, 3$, 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 [35, 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 [19]. 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[10] for the LLL step.

- Assuming that $K \neq \mathbb{Q}(\zeta_5)$, for which the result is known, 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 [29], 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.

7.1 Computation of ϑ-constants

We report timing results for the computation of fundamental ϑ-constants for two arbitrary period matrices. Table 1 shows that already our implementation of the relatively simple naive algorithm presented in §5.1 may be several orders of magnitude faster[3]

[3]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).

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 §5.1 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 §5.1. We note that the timings of Newton lifting depend much less on the 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 1: Computation 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 $O(\mathrm{M}(N)\log N)$. The code in CMH implements the approach using finite differences for estimating the Jacobian matrix as described in §5.4, as well as an algorithm which computes the exact Jacobian matrix along with the Borchardt mean as given in [15, 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.

7.2 Breakdown of timings for small class polynomial examples

Table 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 that altogether we need to carry out 62 lifts of ϑ-constants. Instead of targeting a given precision based on arguments as developed

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

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
$H_1, \hat{H}_2, \hat{H}_3 \in \mathbb{C}[X]$	0.1
$H_1, \hat{H}_2, \hat{H}_3 \in K_0^r[X]$	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
$H_1, \hat{H}_2, \hat{H}_3 \in \mathbb{C}[X]$	0.6
$H_1, \hat{H}_2, \hat{H}_3 \in K_0^r[X]$	1.8 + 2×1.4
check	0.7
Total (incl. I/O)	69.2

Table 2: Timings in seconds for two examples (on one Intel i5-2500, 3.3GHz)

in [33], we simply carry out successive lifting steps until the polynomial reconstruction succeeds. This explains the time needed for failed reconstruction attempts in Table 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 $\hat{H}_2, \hat{H}_3 \in K_0^r[X]$. This can be explained by the relative size of the invariants considered by Streng, see [34, Appendix 3].

The timings indicated as "preparation" and "check" in Table 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 whose Jacobian has complex multiplication by the maximal order of K.

8 A large example

Our currently largest example is $K = \mathbb{Q}[X]/(X^4 + 1357X^2 + 3299)$, containing $K_0 = \mathbb{Q}(\sqrt{1828253})$ of class number 2. Its Shimura class group is $\mathfrak{C} = \mathrm{N}_{\Phi^r}(\mathrm{Cl}_{K^r}) \simeq \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/5004\mathbb{Z}$ of size 20016. On one core of an Intel Core i5-4570 clocked at 3.2 GHz, the structure of the class group is obtained with our Pari/Gp script in roughly one second, while the computation of the period matrices and their symbolic reduction into the fundamental domain $\mathcal{F}_2$ takes 388 s.

The associated ϑ-constants consist of 10008 pairs of complex-conjugate values. For the first ten Newton iterations up to a precision of about 2 000 000 bits, we used 640 cores Intel Xeon X5675 at 3.07 GHz; for the last two iterations, 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 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 §5.4, shows that the effective precision indeed almost doubles in each step as predicted by Theorem 12.

precision	δ	time
2 000	—	0.03
3 986	14	0.01
7 970	2	0.1
15 932	8	0.3
31 862	2	1.0
63 718	6	2.9
127 434	2	8.2
254 858	10	24
509 714	2	61
1 019 416	12	150
2 038 832	0	360
4 077 652	12	940
8 155 302	2	2 100

Table 3: Time for lifting steps for example with $\#\mathfrak{C} = 20\,016$.

The lifting step accounts for a total of about 420 CPU days, but thanks to its easy parallelisation on 160 to 640 cores, it was finished in less than 4 days wall-clock time (including additional overhead for writing intermediate results to disk).

The computation of the floating point polynomials H_1, $\hat{H}_2$ and $\hat{H}_3$ was carried out at a precision of 7 850 071 bits (the lowest lifting precision reached for one of the period matrices). After regrouping complex conjugates, the first step consists of $5 \cdot 10008/2 = 25020$ multiplications of monic polynomials of degree 2 with real coefficients, which can be arbitrarily parallelised; we used a machine with 24 Intel Xeon E7540 cores at 2.0 GHz and 512 GB of memory. At degrees 2 048 to 8192, the FFT multiplications required too much memory to be executed in parallel on all cores, so we reduced the number of simultaneous multiplications to the maximum possible, as indicated in Table 4. In the last step, we needed to multiply a degree 16 384 polynomial with a degree 3 632 polynomial. The wall-clock time of this polynomial reconstruction step, counting input-output time and the cost of resuming computations, was almost exactly 3 days.

Recognising one coefficient of the floating point polynomials as an element of K_0^r took on average 980 s per coefficient on one Intel Xeon E5-2650 core at 2 GHz. The total CPU time for the 60 045 coefficients was thus about 680 CPU days; with up to 480 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 90 GB. The common denominator of the coefficients of H_1 has 8 884 distinct prime factors, the largest one being 1 506 803 839. It occurs to powers 2 in H_1 and 4 in $\hat{H}_2$ and $\hat{H}_3$, 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)	#threads
2	25 020	380	24
4	12 510	560	24
8	6 225	780	24
16	3 125	1 300	24
32	1 565	1 800	24
64	780	2 700	24
128	390	3 700	24
256	195	6 300	24
512	100	11 000	24
1 024	50	13 000	24
2 048	25	25 000	10
4 096	10	20 000	10
8 192	5	71 000	3
16 384	5	67 000	5

Table 4: Polynomial reconstruction timings for example with $\#\mathfrak{C} = 20\,016$.

Acknowledgements

The authors would like to acknowledge the work of Régis Dupont, whose thesis has been extensively used as a basis from the outset of this work. We are also grateful to Damien Robert and Marco Streng for fruitful discussions and to David Gruenewald for his comments on an earlier version.

Computer experiments have used a variety of computing resources funded from different projects. We thus acknowledge the support of the Région Lorraine CPER MISN TALC project; the PlaFRIM experimental testbed, being developed under the Inria PlaFRIM development action with support from LABRI and IMB and other entities: Conseil Régional d'Aquitaine, FeDER, Université de Bordeaux and CNRS (see `https://plafrim.bordeaux.inria.fr/`); the computing facilities MCIA (Mésocentre de Calcul Intensif Aquitain) of the Université de Bordeaux and of the Université de Pau et des Pays de l'Adour.

This research was partially funded by ERC Starting Grant ANTICS 278537 and by Agence Nationale de la Recherche grants ANR-09-BLAN-0020-01 and ANR-12-BS01-0010-01.

References

[1] Karim Belabas et al. PARI/GP. Bordeaux, September 2013. Release 2.5.5, `http://pari.math.u-bordeaux.fr/`.

[2] Juliana Belding, Reinier Bröker, Andreas Enge, and Kristin Lauter. Computing hilbert class polynomials. In Alfred J. van der Poorten and Andreas Stein,

editors, *Algorithmic Number Theory, 8th International Symposium, ANTS-VIII, Banff, Canada, May 17-22, 2008, Proceedings*, volume 5011 of *Lecture Notes in Comput. Sci.*, pages 282–295. Springer–Verlag, 2008.

[3] Carl-Wilhelm Borchardt. Das arithmetisch-geometrische Mittel aus vier Elementen. *Monatsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, pages 611–621, November 1876.

[4] Carl-Wilhelm Borchardt. Theorie des arithmetisch-geometrischen Mittels aus vier Elementen. *Mathematische Abhandlungen der Königlichen Akademie der Wissenschaften zu Berlin*, pages 33–96, 1878.

[5] Jonathan M. Borwein and Peter B. Borwein. *Pi and the AGM.* John Wiley and Sons, 1987.

[6] Richard Brent and Paul Zimmermann. *Modern Computer Arithmetic*, volume 18 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, 2010.

[7] Richard P. Brent. Fast multiple-precision evaluation of elementary functions. *Journal of the ACM*, 23(2):242–251, 1976.

[8] Reinier Bröker. A p-adic algorithm to compute the Hilbert class polynomial. *Mathematics of Computation*, 77(264):2417–2435, 2008.

[9] Reinier Bröker, David Gruenewald, and Kristin Lauter. Explicit CM theory for level 2-structures on abelian surfaces. *Algebra Number Theory*, 5(4):495–528, 2011.

[10] David Cadé, Xavier Pujol, and Damien Stehlé. FPLLL, May 2013. Release 4.0.4, `http://perso.ens-lyon.fr/damien.stehle/fplll/`.

[11] Henri Cohen. *A course in algorithmic algebraic number theory*, volume 138 of *Grad. Texts in Math.* Springer–Verlag, 1993.

[12] Romain Cosset. *Applications des fonctions thêta à la cryptographie sur courbes hyperelliptiques.* Thèse, Université Henri Poincaré - Nancy I, 2011. `http://tel.archives-ouvertes.fr/tel-00642951`.

[13] Jean-Marc Couveignes and Thierry Henocq. Action of modular correspondences around CM points. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory — ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 234–243, Berlin, 2002. Springer-Verlag.

[14] Régis Dupont. Fast evaluation of modular functions using Newton iterations and the AGM. *Mathematics of Computation*, 80(275):1823–1847, 2011.

[15] Régis Dupont. *Moyenne arithmético-géométrique, suites de Borchardt et applications.* Thèse, École Polytechnique, 2006. `http://www.lix.polytechnique.fr/Labo/Regis.Dupont/these_soutenance.pdf`.

[16] Andreas Enge. The complexity of class polynomial computation via floating point approximations. *Mathematics of Computation*, 78(266):1089–1107, 2009.

[17] Andreas Enge. MPFRCX — *A library for univariate polynomials over arbitrary precision real or complex numbers.* INRIA, May 2013. Release 0.4.2, `http://mpfrcx.multiprecision.org/`.

[18] Andreas Enge, Mickaël Gastineau, Philippe Théveny, and Paul Zimmermann. GNU MPC — *A library for multiprecision complex arithmetic with exact rounding.* INRIA, September 2012. Release 1.0.1, `http://mpc.multiprecision.org/`.

[19] Andreas Enge and François Morain. Fast decomposition of polynomials with known Galois group. In Marc Fossorier, Tom Høholdt, and Alain Poli, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes — AAECC-15*, volume 2643 of *Lecture Notes in Computer Science*, page 254–264, Berlin, 2003. Springer-Verlag.

[20] Andreas Enge and Emmanuel Thomé. CMH — *Computation of Igusa Class Polynomials*, December 2013. Development version, `http://cmh.gforge.inria.fr/`.

[21] Pierrick Gaudry, Thomas Houtmann, David R. Kohel, Christophe Ritzenthaler, and Annegret Weng. The 2-adic CM method for genus 2 curves with application to cryptography. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Comput. Sci.*, pages 114–129. Springer-Verlag, 2006.

[22] Erhard Gottschling. Explizite Bestimmung der Randflächen des Fundamentalbereiches der Modulgruppe zweiten Grades. *Math. Ann.*, 138:103–124, 1959.

[23] Torbjörn Granlund et al. GMP — *The* GNU *Multiple Precision Arithmetic Library*, September 2013. Release 5.1.3, `http://gmplib.org/`.

[24] Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, and Paul Zimmermann et al. GNU MPFR — *A library for multiple-precision floating-point computations with exact rounding*, March 2013. Relase 3.1.2, `http://www.mpfr.org/`.

[25] Jun-Ichi Igusa. On Siegel modular forms of genus two. *American Journal of Mathematics*, 84:175–200, 1962.

[26] Jun-Ichi Igusa. *Theta functions*, volume 194 of *Die Grundlehren der mathematischen Wissenschaften.* Springer, 1972.

[27] Helmut Klingen. *Introductory lectures on Siegel modular forms*, volume 20 of *Cambridge studies in advanced mathematics.* Cambridge University Press, 1990.

[28] Kristin E. Lauter and Damien Robert. Improved CRT algorithm for class polynomials in genus 2. In Everett W. Howe and Kiran S. Kedlaya, editors, *ANTS X — Proceedings of the Tenth Algorithmic Number Theory Symposium*, volume 1 of *The Open Book Series*, page 437–461, Berkeley, 2013. Mathematical Sciences Publisher.

[29] Jean-François Mestre. Construction de courbes de genre 2 à partir de leurs modules. In Teo Mora and Carlo Traverso, editors, *Effective methods in algebraic geometry*, volume 94 of *Progr. Math.*, page 313–334. Birkhäuser, 1991.

[30] Goro Shimura. *Abelian Varieties with Complex Multiplication and Modular Functions.* Princeton University Press, 1998.

[31] Goro Shimura and Yutaka Taniyama. *Complex Multiplication of Abelian Varieties and its Applications to Number Theory.* The Mathematical Society of Japan, 1961.

[32] Anne-Monika Spallek. *Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen.* PhD thesis, Universität Gesamthochschule Essen, 1994.

[33] Marco Streng. Computing Igusa class polynomials. To appear in *Mathematics of Computation*, `http://arxiv.org/abs/0903.4766`, 2009.

[34] Marco Streng. *Complex multiplication of abelian surfaces.* Proefschrift, Universiteit Leiden, 2010.

[35] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, Cambridge, England, 1999.