Evaluating modular equations for abelian surfaces

Jean Kieffer

Evaluating modular equations for abelian surfaces

Jean Kieffer

March 3, 2022

Abstract

We design algorithms to efficiently evaluate modular equations of Siegel and Hilbert type for abelian surfaces over number fields using complex approximations. Their output can be made provably correct if an explicit description of the associated graded ring of modular forms over $\mathbb{Z}$ is known; this includes the Siegel case, and the Hilbert case for the quadratic fields of discriminant 5 and 8. Our algorithms also apply to finite fields via lifting.

1 Introduction

Modular equations. Modular equations of Siegel and Hilbert type for abelian surfaces [7, 40, 37, 41] are higher-dimensional analogues of the classical modular polynomials for elliptic curves [11, §11.C]. If ℓ is a prime, then the Siegel modular equations of level ℓ consist of three rational fractions in four variables

$$\Psi_{\ell,1}, \Psi_{\ell,2}, \Psi_{\ell,3} \in \mathbb{Q}(J_1, J_2, J_3)[X],$$

and encode the presence of ℓ-isogenies between principally polarized (p.p.) abelian surfaces. More precisely, let A and A' be two sufficiently generic p.p. abelian surfaces over a field k whose characteristic is prime to ℓ, and denote their Igusa invariants by (j_1, j_2, j_3) and (j_1', j_2', j_3') respectively. Then the equalities

$$\Psi_{\ell,1}(j_1, j_2, j_3, j_1') = 0 \quad \text{and} \quad j_k' = \frac{\Psi_{\ell,k}(j_1, j_2, j_3, j_1')}{\partial_X \Psi_{\ell,1}(j_1, j_2, j_3, j_1')} \quad (k = 2, 3)$$

hold if and only if A and A' are ℓ-isogenous (i.e. related by an isogeny of degree ℓ^2 with isotropic kernel in $A[\ell]$) over an algebraic closure of k.

Similarly, Hilbert modular equations describe certain cyclic isogenies between Jacobians with real multiplication (RM) by a fixed real quadratic field. Modular equations of both Siegel and Hilbert type are examples of the general notion of modular equations on a PEL Shimura variety [29].

From a computational point of view, modular equations are useful to detect isogenies without prior knowledge of their kernels, and also to compute these isogenies explicitly: see [15, 6] in the case of elliptic curves, and [31] in the case of p.p. abelian surfaces. In contrast, all other available methods to compute isogenies, to the author's knowledge, use a description of the kernel as part of their input [56, 10, 36, 13]. This makes modular equations an essential tool in the SEA algorithm for counting points on elliptic curves over finite fields [49], which goes the other way around, using isogenies to gain access to cyclic subgroups of the elliptic curve.

However, modular equations for abelian surfaces are very large objects. One can show that the total size of Siegel modular equations of level ℓ is $O(\ell^{15} \log \ell)$ [29], a prohibitive estimate that we expect to be accurate. Already for $\ell = 3$, their total size is approximately 410 MB [39].

Main results. In this paper, we argue that precomputing modular equations in full is not the correct strategy in higher dimensions. In most contexts, we only need *evaluations* of modular equations, and possibly their derivatives, at a given point $(j_1, j_2, j_3) \in L^3$, where L is a number field; finite fields reduce to this case via lifts. These univariate polynomials can be evaluated directly using complex approximations, building on Dupont's algorithm [14, Chap. 10], [33] to compute genus 2 theta constants in quasi-linear time, now proved to be correct [27]. The resulting algorithm has a much lower asymptotic complexity than precomputing and storing the modular equations in full. For instance, in the case of Siegel modular equations over a prime finite field $\mathbb{F}_p$, we obtain the following result.

Theorem 1.1. *There exists an algorithm which, given prime numbers p and ℓ, and given $(j_1, j_2, j_3) \in \mathbb{F}_p^3$ where the denominator of the Siegel modular equations $\Psi_{\ell,k}$ does not vanish, evaluates the polynomials $\Psi_{\ell,k}(j_1, j_2, j_3)$ and $\partial_{J_i} \Psi_{\ell,k}(j_1, j_2, j_3)$ for $1 \leq i, k \leq 3$ as elements of $\mathbb{F}_p[X]$ within $\widetilde{O}(\ell^6 \log p)$ binary operations.*

A similar results holds for Hilbert modular equations encoding cyclic isogenies of degree ℓ between p.p. abelian surfaces with RM.

Theorem 1.2. *Let F be either $\mathbb{Q}(\sqrt{5})$ or $\mathbb{Q}(\sqrt{8})$; let Δ_F be its discriminant and $\mathbb{Z}_F$ its ring of integers. There exists an algorithm which, given a totally positive prime element $\beta \in \mathbb{Z}_F$ of prime norm $\ell \in \mathbb{Z}$ which is prime to Δ_F, given a prime number p, and given $(g_1, g_2) \in \mathbb{F}_p^2$ where the denominator of the Hilbert modular equations of level β in Gundlach invariant does not vanish, evaluates these modular equations and their derivatives at (g_1, g_2) as elements of $\mathbb{F}_p[X]$ within $\widetilde{O}(\ell^2 \log p)$ binary operations.*

In both cases, we save a factor of $\log p$ when the input values can be written as quotients of small integers: see for instance Theorem 5.4.

The complexity estimate in Theorem 1.1 is small enough to make Elkies's method viable in the context of counting points on abelian surfaces over finite fields [28]. An implementation of the algorithms described in this paper, based on the C libraries Flint [19] and Arb [25], is publicly available [26].

Our algorithm is inspired from existing methods to evaluate elliptic modular polynomials via complex approximations [16]. Note that other methods of based on isogeny graphs [8, 52] allow us to compute or evaluate Φ_ℓ over finite fields with better asymptotic complexities; however, they crucially rely on writing the modular equation in full over small auxiliary finite fields, and it is unclear whether this strategy can still be competitive when the number of variables increases.

Overview of the algorithm. Fix Igusa invariants (j_1, j_2, j_3) over a number field L. For every complex embedding μ of L, we compute a genus 2 hyperelliptic curve $\mathcal{C}$ over $\mathbb{C}$ with Igusa invariants $\big(\mu(j_1), \mu(j_2), \mu(j_3)\big)$. Then, we compute a period matrix τ of $\mathcal{C}$ using AGM sequences [3, 4, 24]. Finally, we compute approximations of $\mu\big(\Psi_{\ell,k}(j_1, j_2, j_3)\big) \in \mathbb{C}[X]$ for $1 \leq k \leq 3$ using analytic formulas. This is done by enumerating carefully chosen period matrices of abelian surfaces that are ℓ-isogenous to $\mathrm{Jac}(\mathcal{C})$, reducing them to a neighborhood of the Siegel fundamental domain, and computing theta constants at these period matrices using Dupont's algorithm. In the case of Hilbert modular equations, an additional step consists in computing an approximate preimage of τ in Hilbert space.

By repeating this whole procedure at increasing precisions, we obtain increasingly better complex approximations of the desired result, a set of

three polynomials in $L[X]$. We know that will eventually be able to correctly recognize its coefficients as algebraic numbers, and a possible strategy is to stop when tentative algebraic reconstructions seem to stabilize. Height bounds on the output [29, Thm. 1.1] suffice to bound the asymptotic complexity of this algorithm. However, its output is difficult to certify when these height bounds are either not explicit or too large for practical use: see for instance [29, Thm. 5.19]. Still, the output is very likely to be correct, and other methods might be available to certify it a posteriori, for instance by certifying the existence of an isogeny between p.p. abelian surfaces [9].

In some cases, we are able to circumvent these problems by reconstructing only algebraic *integers* in L: we separate the numerator and denominator of modular equations using an explicit description of the corresponding graded ring of modular forms with integral Fourier coefficients. As a corollary, we obtain provably correct evaluation algorithms for modular equations of Siegel type, and of Hilbert type for RM discriminants 5 and 8, as announced above.

Organization of the paper. In Section 2, we recall the definition of Siegel and Hilbert modular equations and give explicit formulas for their denominators. In Section 3, we present our computational model and study precision losses in certain polynomial operation appearing in the sequel. Section 4 focuses on computations in the analytic Siegel and Hilbert moduli spaces. In Section 5, we conclude on the cost of the whole algorithm, focusing the case of Hilbert modular equations for $\mathbb{Q}(\sqrt{5})$.

Acknowledgement. This work reproduces part of the author's PhD dissertation at the University of Bordeaux, France. He warmly thanks Damien Robert and Aurel Page for their suggestions, advice, and encouragement.

2 Modular equations for abelian surfaces

In this section, we recall the analytic formulas defining Siegel and Hilbert modular equations for abelian surfaces. We also study their denominators using the structures of the corresponding rings of modular forms over $\mathbb{Z}$.

2.1 Modular equations of Siegel type

Invariants on the Siegel moduli space. Let $\mathcal{H}_2$ be the set of symmetric 2×2 complex matrices τ such that $\operatorname{Im}\tau$ is positive definite. Our notation for the action of the symplectic group $\mathrm{GSp}_4(\mathbb{Q})$ on $\mathcal{H}_2$ is the following: for every $\gamma \in \mathrm{GSp}_4(\mathbb{Q})$ and $\tau \in \mathcal{H}_2$, we write

$$\gamma\tau = (a\tau + b)(c\tau + d)^{-1} \quad \text{and} \quad \gamma^*\tau = c\tau + d,$$

where

$$\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ in } 2 \times 2 \text{ blocks.}$$

We also write $\mathrm{Sp}_4(\mathbb{Z}) = \Gamma(1)$. Recall that the quotient $\Gamma(1)\backslash\mathcal{H}_2$ is a coarse moduli space for p.p. abelian surfaces over $\mathbb{C}$ [1, Thm. 8.2.6], and is the set of complex points of an algebraic variety defined over $\mathbb{Q}$, by the general theory of Shimura varieties [42, §14]. For a subring $R \subset \mathbb{C}$, we denote by $\mathrm{MF}(\Gamma(1), R)$ the graded R-algebra of Siegel modular forms of even weight defined over R, i.e. whose Fourier coefficients are elements of R [55, §4].

The $\mathbb{C}$-algebra $\mathrm{MF}(\Gamma(1), \mathbb{C})$ is free over four generators h_4, h_6, h_{10}, h_{12} [21], where subscripts denote weights. These modular forms can be defined in terms of genus 2 theta constants (of level $(2,2)$), which we now introduce. Let $a, b \in \{0,1\}^2$, seen as vertical vectors. Then the function

$$\theta_{a,b}(\tau) = \sum_{m \in \mathbb{Z}^2} \exp\left(i\pi\left(\left(m + \tfrac{a}{2}\right)^t \tau \left(m + \tfrac{a}{2}\right) + \left(m + \tfrac{a}{2}\right)^t \tfrac{b}{2}\right)\right) \tag{1}$$

is holomorphic on $\mathcal{H}_2$, and is called the theta constant of characteristic (a, b). A traditional indexing, used for instance in [14, 50], is to denote $\theta_{(a_1,a_2),(b_1,b_2)}$ as θ_j where $j = 8b_2 + 4b_1 + 2a_2 + a_1$ is an integer between 0 and 15. The only nonzero theta constants are the *even* ones, for which the dot product $a^t b$ is even; they are indexed by $j \in \{0, 1, 2, 3, 4, 6, 8, 9, 12, 15\}$.

It is known that every Siegel modular form in dimension 2 is a polynomial in theta constants [22], and the explicit expressions of $h_4, \ldots, h_{12}$ can be found in [50, (7.1) p. 68]. In particular, h_{10} is the product of all squares of even theta constants, and is therefore a scalar multiple of the traditional cusp form χ_{10}. Therefore $\tau \in \mathcal{H}_2$ is the period matrix of a genus 2 hyperelliptic curve over $\mathbb{C}$ if and only if $h_{10}(\tau) \neq 0$.

Following [50, §2.1], we define the Igusa invariants as follows:

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

They realize a birational map between $\Gamma(1)\backslash\mathcal{H}_2$ and the projective space $\mathbb{P}^3$. This map is defined over $\mathbb{Q}$ by the q-expansion principle [17, §V.1.5]. Therefore, the Igusa invariants (2) are suitable coordinates in which modular equations for p.p. abelian surfaces can be written.

We will also use the structure of $\mathrm{MF}(\Gamma(1),\mathbb{Z})$. Igusa [23] computed fourteen explicit generators for this ring. For our purposes, it is sufficient to note that $\mathbb{Z}[h_4,h_6,h_{10},h_{12}]\subset\mathrm{MF}(\Gamma(1),\mathbb{Z})$ is a reasonably large subring.

Lemma 2.1. *Let $f\in\mathrm{MF}(\Gamma(1),\mathbb{Z})$ be a modular form of weight w. Then*

$$2^{\lfloor 7w/4\rfloor}3^{\lfloor w/4\rfloor}f\in\mathbb{Z}[h_4,h_6,h_{10},h_{12}].$$

Proof. The lowest weight generators of $\mathrm{MF}(\Gamma(1),\mathbb{Z})$ are

$$X_4=2^{-2}h_4,\quad X_6=2^{-2}h_6,\quad X_{10}=-2^{-12}h_{10},\quad X_{12}=2^{-15}h_{12}.$$

The result holds for these four generators. Direct computations using the formulas from [23, p. 153] show that it also holds for the ten others. □

Proposition 2.2. *Let $f\in\mathrm{MF}(\Gamma(1),\mathbb{Z})$ be a modular form of weight w. Let $a\geq 0$ and $0\leq b\leq 4$ be integers such that $4\lfloor w/6\rfloor+w=10a+4b$. Then there exists a unique polynomial $Q_f\in\mathbb{Z}[J_1,J_2,J_3]$ such that the following equality of Siegel modular functions holds:*

$$Q_f(j_1,j_2,j_3)=2^{\lfloor 7w/4\rfloor}3^{\lfloor w/4\rfloor}\frac{h_4^{\lfloor w/6\rfloor}}{h_{10}^a h_4^b}f.$$

The degree of Q_f in J_1,J_2,J_3 is bounded above by $\lfloor w/6\rfloor,\lfloor w/12\rfloor,\lfloor w/12\rfloor$ respectively, and the total degree of Q_f is bounded above by $\lfloor w/6\rfloor$.

Proof. By Lemma 2.1, we can rewrite the right hand side as

$$\frac{h_4^{\lfloor w/6\rfloor}}{h_{10}^b h_4^a}F$$

for some $F\in\mathbb{Z}[h_4,h_6,h_{10},h_{12}]$. Then the equalities

$$h_4h_6=j_1h_{10},\quad h_4^2h_{12}=j_2h_{10}^2,\quad h_4^5=j_3h_{10}^2$$

show that this quotient can be rewritten as a polynomial in j_1,j_2,j_3 with integer coefficients satisfying the required degree bounds, as detailed in [29, Lem. 4.7]. The resulting polynomial Q_f is unique because j_1,j_2,j_3 are algebraically independent. □

Siegel modular equations. Let $\ell \in \mathbb{Z}$ be a prime. We define the subgroup $\Gamma^0(\ell)$ of $\Gamma(1) = \mathrm{Sp}_4(\mathbb{Z})$ as follows:

$$\Gamma^0(\ell) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma(1) \colon b = 0 \bmod \ell \right\}.$$

Its index in $\Gamma(1)$ is $\ell^3 + \ell^2 + \ell + 1$, and an explicit set C_ℓ of representatives for the quotient $\Gamma^0(\ell)\backslash\Gamma(1)$ is obtained by conjugating the matrices listed in [14, Def. 10.1] by $\left(\begin{smallmatrix} 0 & I_2 \\ -I_2 & 0 \end{smallmatrix}\right)$. The absolute values of all their coefficients are bounded above by ℓ.

The Siegel modular equations of level ℓ [7, 40] are the three multivariate rational fractions

$$\Psi_{\ell,k} \in \mathbb{Q}(J_1, J_2, J_3)[X], \quad 1 \le k \le 3$$

such that for every $\tau \in \mathcal{H}_2$ where everything is well defined, we have

$$\begin{aligned}
\Psi_{\ell,1}\big(j_1(\tau), j_2(\tau), j_3(\tau)\big) &= \prod_{\gamma \in C_\ell} \big(X - j_1(\tfrac{1}{\ell}\gamma\tau)\big), \\
\Psi_{\ell,2}\big(j_1(\tau), j_2(\tau), j_3(\tau)\big) &= \sum_{\gamma \in C_\ell} j_2(\tfrac{1}{\ell}\gamma\tau) \prod_{\gamma' \in C_\ell \setminus \{\gamma\}} \big(X - j_1(\tfrac{1}{\ell}\gamma'\tau)\big), \\
\Psi_{\ell,3}\big(j_1(\tau), j_2(\tau), j_3(\tau)\big) &= \sum_{\gamma \in C_\ell} j_3(\tfrac{1}{\ell}\gamma\tau) \prod_{\gamma' \in C_\ell \setminus \{\gamma\}} \big(X - j_1(\tfrac{1}{\ell}\gamma'\tau)\big).
\end{aligned} \tag{3}$$

The degrees of the polynomials $\Psi_{\ell,k}$ in X are at most $\ell^3+\ell^2+\ell+1$, and their total degrees in J_1, J_2, J_3 are at most $10(\ell^3+\ell^2+\ell+1)/3$ by [29, Prop. 4.10]. The height of their coefficients is $O(\ell^3 \log \ell)$ by [29, Thm. 1.1].

Change of representatives. The analytic formulas (3), combined with the expression (2) of Igusa invariants, show that modular equations of Siegel type can be evaluated at some $\tau \in \mathcal{H}_2$ by computing theta constants at period matrices of the form $\gamma\tau$, where

$$\gamma \in \begin{pmatrix} I_2 & 0 \\ 0 & \ell I_2 \end{pmatrix} C_\ell \subset \mathrm{GSp}_4(\mathbb{Q}). \tag{4}$$

We now show how to modify these matrices γ so that their lower left block becomes zero. This will be essential to achieve good control on the complexity of numerical computations later on, as it implies that $\gamma\tau$ will not be too far

from the fundamental domain when τ satisfies the same property. Similarly, in the dimension 1 case, $\mathrm{SL}_2(\mathbb{Z})$ acts on the upper half plane $\mathcal{H}_1$, and matrices leaving the cusp at infinity invariant are precisely the upper triangular ones.

Proposition 2.3. *Let $\gamma \in \mathrm{GSp}_4(\mathbb{Q})$ be a symplectic matrix with integer coefficients. Then there exists a matrix $\eta \in \mathrm{Sp}_4(\mathbb{Z})$ such that $\eta\gamma$ has a zero lower left 2×2 block. A suitable η such that $\log|\eta| = O(\log|\gamma|)$ can be computed in $\widetilde{O}(\log|\gamma|)$ binary operations.*

Proof. Let a, b, c, d be the 2×2 blocks of γ. First, we choose the two lower lines of η as a basis (l_3, l_4) of $V \cap \mathbb{Z}^4$, where V is the 2-dimensional $\mathbb{Q}$-vector space

$$V = \mathbb{Q}^2 \begin{pmatrix} -c & a \end{pmatrix}.$$

This ensures that the lower left block of $\eta\gamma$ is zero. Since γ is symplectic, V is an isotropic subspace of $\mathbb{Q}^4$ for the standard symplectic pairing, denoted by $\langle \cdot, \cdot \rangle$. Therefore, it is possible to complete (l_3, l_4) in a symplectic basis $(l_1, \ldots, l_4)$ of $\mathbb{Z}^4$, providing the required $\eta \in \mathrm{Sp}_4(\mathbb{Z})$.

The lines l_3 and l_4 can be obtained from the classical algorithm for elementary divisors over $\mathbb{Z}$, giving $\log|l_k| = O(\log|\gamma|)$ for $k = 3, 4$; this costs $\widetilde{O}(\log|\gamma|)$ binary operations. Choose any l_1, l_2 with coefficients in $\mathbb{Z}$ such that $\langle l_1, l_3 \rangle = \langle l_2, l_4 \rangle = 1$ and $\langle l_1, l_4 \rangle = \langle l_2, l_3 \rangle = 0$; this can be done using the Euclidean algorithm, and defines l_1, l_2 uniquely up to translations by $\mathbb{Z}l_3 \oplus \mathbb{Z}l_4$. Note that

$$\langle l_1 + xl_3 + yl_4, l_2 + zl_3 + tl_4 \rangle = \langle l_1, l_2 \rangle + z - y,$$

so l_1, l_2 can easily be corrected to ensure that $\langle l_1, l_2 \rangle = 0$. The final basis satisfies $\log|l_k| = O(\log|\gamma|)$ for $1 \leq k \leq 4$. □

Applying Proposition 2.3 to the matrices listed in (4), we obtain a collection D_ℓ of $\ell^3 + \ell^2 + \ell + 1$ matrices $\gamma \in \mathrm{GSp}_4(\mathbb{Q})$ with integer coefficients, lower left block zero, determinant ℓ^2, and such that $\log|\gamma| = O(\log \ell)$. They can be used to rewrite the formulas (3): for instance,

$$\Psi_{\ell,1}\big(j_1(\tau), j_2(\tau), j_3(\tau)\big) = \prod_{\gamma \in D_\ell} \big(X - j_1(\gamma\tau)\big).$$

Denominators. We call a polynomial $Q_\ell \in \mathbb{Z}[J_1, J_2, J_3]$ a *denominator* of the Siegel modular equations $\Psi_{\ell,k}$ if for each $1 \leq k \leq 3$, we have

$$Q_\ell \Psi_{\ell,k} \in \mathbb{Z}[J_1, J_2, J_3, X].$$

Our goal is describe such a denominator by an analytic formula. For every $\tau \in \mathcal{H}_2$, we define

$$g_\ell(\tau) = \ell^{-20(2\ell^2+\ell+1)} \prod_{\gamma \in \Gamma^0(\ell)\backslash\Gamma(1)} \det(\gamma^*\tau)^{-20} 2^{-24} h_{10}^2(\tfrac{1}{\ell}\gamma\tau). \tag{5}$$

One can check that g_ℓ is well-defined Siegel modular form of weight

$$w_\ell = 20(\ell^3 + \ell^2 + \ell + 1).$$

Using the set of matrices D_ℓ defined above, we can also write

$$g_\ell(\tau) = \ell^{-20(2\ell^2+\ell+1)} \prod_{\gamma \in D_\ell} (\ell^2 \det(\gamma^*\tau)^{-1})^{20} 2^{-24} h_{10}^2(\gamma\tau).$$

In this product, $\det(\gamma^*\tau)$ is simply the determinant of the lower right block of γ, and is independent of τ. Since γ has integer coefficients and determinant ℓ^2, we see that $\ell^2 \det(\gamma^*\tau)^{-1}$ is either ± 1, $\pm\ell$, or $\pm\ell^2$. After computing D_ℓ explicitly, we observe that ± 1 happens ℓ^3 times, $\pm\ell$ happens $\ell + 1$ times, and $\pm\ell^2$ happens ℓ^2 times; therefore,

$$g_\ell(\tau) = \prod_{\gamma \in D_\ell} 2^{-24} h_{10}^2(\gamma\tau). \tag{6}$$

This cancellation justifies the introduction of a power of ℓ in (5).

For every $0 \leq i \leq \ell^3 + \ell^2 + \ell + 1$ and $1 \leq k \leq 3$, we also define $f_{\ell,k,i}(\tau)$ as the coefficient of X^i in the polynomial $g_\ell(\tau)\Psi_{\ell,k}\big(j_1(\tau), j_2(\tau), j_3(\tau)\big)$.

Proposition 2.4. *The functions g_ℓ and $f_{\ell,k,i}$ are Siegel modular forms of weight w_ℓ defined over $\mathbb{Z}$.*

Proof. The modular form g_ℓ has an algebraic interpretation: up to a rational scalar factor, it is the image of the modular form h_{10}^2 under the Hecke correspondence associated with ℓ-isogenies of abelian surfaces [17, §VII.3]. Therefore, the modular form g_ℓ is defined over $\mathbb{Q}$, and so are the modular forms $f_{\ell,k,i}$, since each $f_{\ell,k,i}/g_\ell$ is a modular function defined over $\mathbb{Q}$.

Let $\gamma \in D_\ell$. By writing down the Fourier expansion of h_{10}, we see that the function $\tau \mapsto 2^{-24} h_{10}^2(\gamma\tau)$ has a Fourier expansion in terms of the entries of τ/ℓ^2 with coefficients in $\mathbb{Z}[\exp(2\pi i/\ell^2)]$. Since g_ℓ is a product of such functions by (6), this shows that the Fourier coefficients of g_ℓ are (algebraic) integers. Similarly, the modular forms $f_{\ell,k,i}$ are sums of products of such functions, so their Fourier coefficients are integers. □

Finally, let Q_ℓ be the polynomial defined in Proposition 2.2 applied to the modular form g_ℓ.

Theorem 2.5. *The polynomial $Q_\ell \in \mathbb{Z}[J_1, J_2, J_3]$ is a denominator of the Siegel modular equations of level ℓ.*

Proof. For each $1 \le k \le 3$ and $0 \le i \le \ell^3 + \ell^2 + \ell + 1$, the coefficient of X^i in the rational fraction $Q_\ell \Psi_{\ell,k} \in \mathbb{Q}(J_1, J_2, J_3)[X]$ is precisely the polynomial $Q_{f_{\ell,k,i}}$ from Proposition 2.2. □

For each $\tau \in \mathcal{H}_2$ such that $h_{10}(\tau) \neq 0$, unfolding the definitions provides explicit formulas to evaluate Q_ℓ and $Q_\ell \Psi_{\ell,k}$ for $1 \le k \le 3$ at the point $\big(j_1(\tau), j_2(\tau), j_3(\tau)\big)$ given the values of theta constants at $\gamma\tau$ for $\gamma \in D_\ell$.

2.2 Modular equations of Hilbert type

Invariants on Hilbert moduli spaces. Fix a real quadratic field F of discriminant Δ_F, and let $\mathbb{Z}_F$ be its ring of integers. We fix a real embedding of F, and denote the other embedding by $x \mapsto \overline{x}$. Then $\mathrm{GL}_2(F)$ embeds in $\mathrm{GL}_2(\mathbb{R})^2$ via the two real embeddings of F, so has a natural action on $\mathcal{H}_1^2$, denoted by $(\gamma, t) \mapsto \gamma t$. For every $\gamma \in \mathrm{GL}_2(F)$, every $\alpha \in F$ and every $t = (t_1, t_2) \in \mathcal{H}_1^2$, we also write

$$\alpha t = (\alpha t_1, \overline{\alpha} t_2), \quad \text{and} \quad \gamma^* t = (ct_1 + d)(\overline{c} t_2 + \overline{d}) \quad \text{if } \gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Let $\mathbb{Z}_F^\vee = (1/\sqrt{\Delta_F})\,\mathbb{Z}_F$ be the dual of $\mathbb{Z}_F$ with respect to the trace form, and define the group

$$\Gamma_F(1) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(F) \colon a, d \in \mathbb{Z}_F,\ b \in \mathbb{Z}_F^\vee,\ c \in (\mathbb{Z}_F^\vee)^{-1} \right\}.$$

The quotient $\Gamma_F(1)\backslash\mathcal{H}_1^2$ is a coarse moduli space for p.p. abelian surfaces over $\mathbb{C}$ with real multiplication by $\mathbb{Z}_F$ [1, §9.2]. The involution given by

$$\sigma\colon (t_1, t_2) \mapsto (t_2, t_1)$$

exchanges the real multiplication embedding with its Galois conjugate. As in the Siegel case, $\Gamma_F(1)\backslash\mathcal{H}_1^2$ has a canonical algebraic model which is defined over $\mathbb{Q}$ [54, Chap. X, Rem. 4.1].

For a subring $R \subset \mathbb{C}$, we denote by $\mathrm{MF}(\Gamma_F(1), R)$ the graded R-algebra of Hilbert modular forms of even weight defined over R that are symmetric, i.e. invariant under σ. There is no known general description of these graded algebras, although $\mathrm{MF}(\Gamma_F(1), \mathbb{Z})$ is known for discriminants 5 and 8 [45], $\mathrm{MF}(\Gamma_F(1), \mathbb{Q})$ is known for the additional discriminants $12, 13, 17, 24, 29, 37$ (see [57] and the references therein), and in general they are amenable to computation for a fixed F [12].

One way of defining coordinates on Hilbert moduli spaces consists in pulling back invariants from the Siegel moduli space by the forgetful map. Let (e_1, e_2) be the $\mathbb{Z}$-basis of $\mathbb{Z}_F$; for the sake of fixing definitions, we take $(1, \frac{1}{2}\sqrt{\Delta_F})$ if Δ_F is even, and $(1, \frac{1}{2} + \frac{1}{2}\sqrt{\Delta_F})$ if Δ_F is odd. Write

$$R_F = \begin{pmatrix} e_1 & e_2 \\ \overline{e}_1 & \overline{e}_2 \end{pmatrix} \in \mathrm{GL}_2(\mathbb{R}).$$

Then the *Hilbert embedding* [54, p. 209] is the map

$$\begin{array}{rccc} \Phi_F\colon & \mathcal{H}_1^2 & \to & \mathcal{H}_2 \\ & (t_1, t_2) & \mapsto & R_F^t \begin{pmatrix} t_1 & 0 \\ 0 & t_2 \end{pmatrix} R_F. \end{array} \tag{7}$$

It is equivariant for the actions of $\mathrm{GL}_2(F)$ and $\mathrm{GSp}_4(\mathbb{Q})$ under the following map, also denoted by Φ_F:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto \begin{pmatrix} R_F^t & 0 \\ 0 & R_F^{-1} \end{pmatrix} \begin{pmatrix} a^* & b^* \\ c^* & d^* \end{pmatrix} \begin{pmatrix} R_F^{-t} & 0 \\ 0 & R_F \end{pmatrix} \tag{8}$$

where $x^* = \left(\begin{smallmatrix} x & 0 \\ 0 & \overline{x} \end{smallmatrix}\right)$ for $x \in F$, and R_F^{-t} denotes the inverse of R_F^t.

One can check that Φ_F induces a map $\overline{\Phi}_F\colon \Gamma_F(1)\backslash\mathcal{H}_1^2 \to \Gamma(1)\backslash\mathcal{H}_2$; in the modular interpretation, $\overline{\Phi}_F$ forgets the real multiplication embedding. Generically, the map $\overline{\Phi}_F$ is 2-1: for every $t \in \mathcal{H}_1^2$, we have $\overline{\Phi}_F(t) = \overline{\Phi}_F(\sigma(t))$.

Therefore, the pullback of Igusa invariants under $\overline{\Phi}_F$ can be used to write down modular equations of Hilbert type symmetrized under σ.

When the structure of $\mathrm{MF}(\Gamma_F(1),\mathbb{Z})$ is known, it is better to design other coordinates in terms of a generating set of modular forms. We will focus on the case of $F=\mathbb{Q}(\sqrt{5})$. Then $\mathrm{MF}(\Gamma_F(1),\mathbb{Z})$ is generated by four elements G_2, F_6, F_{10}, and $F_{12}=\frac{1}{4}(F_6^2-G_2F_{10})$, where subscripts denote weights [45]. Following [41], we define the Gundlach invariants g_1, g_2 as

$$g_1=\frac{G_2^5}{F_{10}} \quad \text{and} \quad g_2=\frac{G_2^2F_6}{F_{10}}.$$

The expressions of the pullbacks of $h_4,\dots,h_{12}$ by the Hilbert embedding Φ_F in terms of G_2, F_6, F_{10} can be found in [48]; in particular $\Phi_F^*(h_{10})=2^{12}F_{10}$. As a byproduct, Gundlach and Igusa invariants are related by explicit polynomial formulas [48, Thm. 1], [34, Prop. 4.5].

From the description of $\mathrm{MF}(\Gamma_F(1),\mathbb{Z})$, we immediately have the following analogues of Lemma 2.1 and Proposition 2.2.

Lemma 2.6. *Let $F=\mathbb{Q}(\sqrt{5})$. Then for every $f\in\mathrm{MF}(\Gamma_F(1),\mathbb{Z})$ of weight w, we have $2^{\lfloor w/3\rfloor}f\in\mathbb{Z}[G_2,G_6,F_{10}]$.*

Proposition 2.7. *Let $F=\mathbb{Q}(\sqrt{5})$. Then for every $f\in\mathrm{MF}(\Gamma_F(1),\mathbb{Z})$ of weight w, we have $2^{\lfloor w/3\rfloor}f\in\mathbb{Z}[G_2,G_6,F_{10}]$. Let $a\geq 0$ and $0\leq b\leq 4$ be the unique integers such that $4\lfloor w/6\rfloor+w=10a+2b$. Then there exists a unique polynomial $Q_f\in\mathbb{Z}[J_1,J_2]$ such that the following equality of Hilbert modular function holds:*

$$Q_f(g_1,g_2)=2^{\lfloor w/3\rfloor}\frac{G_2^{2\lfloor w/6\rfloor}}{F_{10}^aG_2^b}f.$$

The total degree of Q_f is bounded above by $\lfloor w/6\rfloor$.

Hilbert modular equations. We fix $F=\mathbb{Q}(\sqrt{5})$ in the rest of this section, and consider modular equations in Gundlach invariants; the formulas in the general case of Hilbert modular equations in Igusa invariants are similar.

Let ℓ be a prime not dividing Δ_F that splits in F in two principal ideals generated by totally positive elements $\beta,\overline{\beta}\in\mathbb{Z}_F$. Define the subgroup $\Gamma_F^0(\beta)$ of $\Gamma_F(1)$ by

$$\Gamma_F^0(\beta)=\left\{\begin{pmatrix}a & b\\ c & d\end{pmatrix}\in\Gamma_F(1)\colon b=0 \bmod \beta\right\}.$$

A set C_β of representatives for the quotient $\Gamma_F^0(\beta)\backslash\Gamma(1)$ consists of the $\ell+1$ following matrices:

$$\begin{pmatrix} 0 & 1/\sqrt{\Delta_F} \\ -\sqrt{\Delta_F} & 0 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 1 & a/\sqrt{\Delta_F} \\ 0 & 1 \end{pmatrix} \quad \text{for } a \in [\![0, \ell-1]\!].$$

A set of representatives for the quotient $\Gamma_F^0(\beta)\backslash(\Gamma(1) \rtimes \langle\sigma\rangle)$ is $C_\beta^\sigma = C_\beta \cup C_\beta\sigma$. The Hilbert modular equations of level β in Gundlach invariants [37, 41] are the two multivariate rational fractions

$$\Psi_{\beta,k} \in \mathbb{Q}(J_1, J_2)[X], \quad 1 \le k \le 2$$

such that for every $t \in \mathcal{H}_1^2$, we have

$$\begin{aligned} \Psi_{\beta,1}\big(g_1(t), g_2(t)\big) &= \prod_{\gamma \in C_\beta^\sigma} \big(X - g_1(\tfrac{1}{\beta}\gamma t)\big), \\ \Psi_{\beta,2}\big(g_1(t), g_2(t)\big) &= \sum_{\gamma \in C_\beta^\sigma} g_2(\tfrac{1}{\beta}\gamma t) \prod_{\gamma' \in C_\beta^\sigma \setminus \{\gamma\}} \big(X - g_1(\tfrac{1}{\beta}\gamma' t)\big). \end{aligned} \tag{9}$$

The degrees of the polynomials $\Psi_{\beta,k}$ in X are at most $2\ell+2$, and their total degrees in g_1, g_2 are at most $10(\ell+1)/3$ by [29, Prop. 4.11]. The height of their coefficients is $O(\ell \log \ell)$ by [29, Thm. 1.1]. In the modular interpretation, these modular equations encode the presence of β-isogenies [13], of degree ℓ, between p.p. abelian surfaces with real multiplication by $\mathbb{Z}_F$.

The formulas (9) show that Hilbert modular equations can be evaluated at $\big(g_1(t), g_2(t)\big)$ for $t \in \mathcal{H}_1^2$ by computing theta constants at each of the period matrices $\Phi_F(\frac{1}{\beta}\gamma t)$ in Siegel space, for $\gamma \in C_\beta^\sigma$. As in the Siegel case, we will let suitable elements of $\Gamma(1)$ act on these period matrices to have better control on the numerical computations; in the Hilbert case, they will be computed during the execution of the algorithm.

As above, we call $Q_\beta \in \mathbb{Z}[J_1, J_2]$ a *denominator* of the Hilbert modular equations $\Psi_{\beta,k}$ if for each $1 \le k \le 2$, we have

$$Q_\beta \Psi_{\beta,k} \in \mathbb{Z}[J_1, J_2].$$

For every $t \in \mathcal{H}_1^2$, we write $g_\beta(t) = f_\beta(t) f_\beta(\sigma(t))$, where

$$f_\beta(t) = \frac{1}{\ell} \prod_{\gamma \in \Gamma_F^0(\beta)\backslash\Gamma_F(1)} (\gamma^* t)^{-10} 2^{-12} h_{10}\big(\Phi_F(\tfrac{1}{\beta}\gamma t)\big). \tag{10}$$

We can check that g_β is a well-defined Hilbert modular form, symmetric and integral, of weight

$$w_\beta = 20(2\ell + 2),$$

and that the polynomial Q_β obtained from Proposition 2.7 applied to g_β is a denominator of the Hilbert modular equations.

3 Precision losses in polynomial operations

In all algorithms manipulating complex numbers, we use ball arithmetic [53]. Given $z \in \mathbb{C}$ and $N \geq 0$, we define an *approximation of z to precision N* to be a complex ball of radius 2^{-N} containing z. An approximation of a polynomial to precision N is by definition an approximation to precision N coefficient per coefficient. Approximations centered at dyadic points can be stored in a computer, and allow us to design algorithms with meaningful input and provably correct output.

If an algorithm takes approximations to precision $N \geq M$ as input and outputs approximations to precision $N - M$ for some $M \geq 0$, we say that the *precision loss* in this algorithm is M bits. For instance, precision losses in elementary operations (additions, multiplications, etc.) can be bounded above in terms of the size of the operands. Besides these theoretical bounds, precision losses can also be computed on the fly in a precise way; this is done in the Arb library [25], on which our implementation is based. If we run out of precision during the computation, we can simply double the precision and restart. Therefore, in the theoretical analysis presented here, it is sufficient to bound the precision losses from above in the O notation.

We let $\mathcal{M}(N)$ be a quasi-linear, superlinear function such that two N-bit integers can be multiplied in $\mathcal{M}(N)$ binary operations. We write log (resp. $\log_2$) for the natural logarithm (resp. logarithm in base 2), and for $x \in \mathbb{R}$, we define

$$\log^+ x = \log \max\{1, x\} \quad \text{and} \quad \log_2^+ x = \log_2 \max\{1, x\}.$$

We denote the modulus of the largest coefficient in a polynomial P by $|P|$; we also use this notation for vectors and matrices.

3.1 Product trees and interpolation

We start with a technical lemma that we will use several times, when we construct polynomials as products of linear factors.

Lemma 3.1. *There exists an algorithm such that the following holds. Let $d \geq 1$, $B \geq 1$, $C \geq 1$, and let x_i, y_i, z_i for $1 \leq i \leq d$ be complex numbers such that*

$$\log^+|x_i| \leq B, \quad \log^+|y_i| \leq B, \quad \log^+|z_i| \leq C, \quad \text{for all } i.$$

Let $N \geq 1$. Then, given approximations of these complex numbers to precision N, the algorithm computes the polynomials

$$P = \prod_{i=1}^{d}(x_i X + y_i) \qquad Q = \sum_{i=1}^{d} z_i \prod_{j \neq i}(x_j X + y_j)$$

within $O\big(\mathcal{M}\big(d(N + C + dB)\big) \log d\big)$ binary operations, with a precision loss of $O(C + dB)$ bits.

Proof. We use product trees [5, §I.5.4]. For each $0 \leq m \leq \lceil \log_2(d) \rceil$, the m-th level of the product tree to compute P consists of $2^{\lceil \log_2(d) \rceil - m}$ products of (at most) 2^m factors of the form $x_i X + y_i$. Hence, for every polynomial R appearing at the m-th level, we have

$$\deg(R) \leq 2^m \quad \text{and} \quad \log^+|R| = O(2^m B).$$

Level 0 is given as input. In order to compute level $m+1$ from level m, we compute one product per vertex, for a total cost of $O\big(\mathcal{M}\big(d(N+dB)\big)\big)$ binary operations; the precision loss in this operation is $O(2^m B)$ bits. Therefore the total precision loss when computing P is $O(dB)$ bits. The number of levels is $O(\log d)$, so the total cost is $O\big(\mathcal{M}\big(d(N + dB)\big) \log d\big)$ binary operations.

For the polynomial Q, the following product tree is used. Each vertex at level $m + 1$ is a polynomial of the form $N_1 P_2 + N_2 P_1$ where P_i is a vertex of the product tree for P satisfying

$$\deg(P_i) \leq 2^m \quad \text{and} \quad \log^+|P_i| = O(2^m B),$$

and the polynomials N_i come from the m-th level, and satisfy

$$\deg(N_i) \leq 2^m - 1 \quad \text{and} \quad \log^+|N_i| = O(C + 2^m B).$$

By induction, the m-th level can be computed to precision $N - O(C + 2^m B)$ using a total of $O\big(\mathcal{M}\big(d(N + C + dB)\big)\big)$ binary operations. □

We apply Lemma 3.1 to Lagrange interpolation.

Proposition 3.2. *There exists an algorithm such that the following holds. Let $P \in \mathbb{Z}[X]$ be an irreducible polynomial of degree $d \geq 1$, let $(\alpha_i)_{1\leq i\leq d}$ be the roots of P, and let $(t_i)_{1\leq i\leq d}$ be complex numbers. Let $M, C \geq 1$ such that*

$$\log^+|P| \leq M, \quad \text{and} \quad \log^+|t_i| \leq C \quad \text{for every } i.$$

Let $N \geq 1$. Then, given P and approximations of the α_i, t_i, and $1/P'(\alpha_i)$ to precision N, the algorithm computes the polynomial Q of degree at most $d-1$ interpolating the points (α_i, t_i) within

$$O\big(\mathcal{M}\big(d(N + C + dM + d\log d)\big)\log d\big)$$

binary operations. The precision loss is $O(C + dM + d\log d)$ bits.

Proof. We write

$$Q = \sum_{i=1}^{d} \frac{t_i}{P'(\alpha_i)} \prod_{j\neq i}(X - \alpha_j).$$

We have $\log^+|P'| \leq M + \log d$. The discriminant $\mathrm{Disc}(P)$ of P is the resultant of P and P'. Hence we can write

$$UP + VP' = \mathrm{Disc}(P)$$

with $U, V \in \mathbb{Z}[X]$; the coefficients of U, V have expressions as determinants of size $O(d)$ involving the coefficients of P and P', so by Hadamard's lemma, we have in particular

$$\log^+|V| = O(dM + d\log d).$$

We have $\log^+|\alpha_i| \leq M + \log(2)$ for every i, hence

$$\log^+\Big|\frac{1}{P'(\alpha_i)}\Big| = \log^+\Big|\frac{V(\alpha_i)}{\mathrm{Disc}(P)}\Big| = O(dM + d\log d).$$

Therefore the precision loss taken when computing the d complex numbers $z_i = t_i/P'(\alpha_i)$ is $O(C + dM + d\log d)$ bits; the total cost to compute the z_i is

$$O\big(d\mathcal{M}(N + C + dM + d\log d)\big)$$

binary operations. We conclude using Lemma 3.1. ∎

3.2 Recognizing integers in number fields

Our goal is now to estimate the necessary precision to recognize integers in a number field L. We give two results according to the description of the number field. In the first description, the number field is $\mathbb{Q}(\alpha)$ where α is a root of some polynomial $P \in \mathbb{Z}[X]$ with bounded coefficients, and we want to recognize an element $x \in \mathbb{Z}[\alpha]$. This situation arises for instance when lifting from a finite field; not much is known about the number field itself. In the second description, we assume that an HKZ- or LLL-reduced basis of the ring of integers $\mathbb{Z}_L$ is known, and we want to recognize an element $x \in \mathbb{Z}_L$. The necessary precision is given in terms of the discriminant Δ_L of L and the absolute logarithmic height $h(x)$ of x, as defined in [20, §B.2].

Proposition 3.3. *There exist an algorithm and an absolute constant C such that the following holds. Let L be a number field of degree d over $\mathbb{Q}$ defined by a monic irreducible polynomial $P \in \mathbb{Z}[X]$, and let $M \geq 1$ such that $\log^+|P| \leq M$. Let α be a root of P in L. Let*

$$x = \sum_{j=0}^{d-1} \lambda_j \alpha^j \in \mathbb{Z}[\alpha]$$

with $\lambda_j \in \mathbb{Z}$ and $\log^+|\lambda_j| \leq H$ for every j. Let $N \geq C(H + dM + d\log d)$. Then, given P and approximations of x, α and $1/P'(\alpha)$ to precision N in every complex embedding of L, the algorithm computes x within

$$O\big(\mathcal{M}\big(d(H + dM + d\log d)\big)\log d\big)$$

binary operations.

Proof. Denote the complex embeddings of L by $\mu_1, \ldots, \mu_d$. The polynomial $Q = \sum_{j=0}^{d-1} \lambda_j X^j$ interpolates the points $(\mu_i(\alpha), \mu_i(x))$ for every $1 \leq i \leq d$. By assumption, we have for each i

$$\log^+|\mu_i(x)| \leq H + O(dM).$$

We are in the situation of Proposition 3.2: we can compute an approximation of Q with a precision loss of $O(H + dM + d\log d)$ bits. Therefore, for an appropriate choice of the constant C, the resulting precision is sufficient to obtain Q exactly by rounding the result to the nearest integers. □

Let L be a number field of degree d over $\mathbb{Q}$. We endow $\mathbb{Z}_L$ with the euclidean metric induced by the map $\mathbb{Z}_L \to \mathbb{C}^d$ given by the d complex embeddings $\mu_1, \dots, \mu_d$ of L. Then $\mathbb{Z}_L$ becomes a lattice of volume Δ_L in the Euclidean space $\mathbb{Z}_L \otimes_{\mathbb{Z}} \mathbb{R}$. Denote by $1 \leq \lambda_1 \leq \cdots \leq \lambda_d$ the successive minima of $\mathbb{Z}_L$. They satisfy the following inequality [46, Chap. 2, Thm. 5]:

$$\prod_{k=1}^{d} \lambda_k \leq \alpha_d^{d/2} \Delta_L,$$

where $\alpha_d \leq 1 + \frac{d}{4}$ denotes Hermite's constant [46, Chap. 2, Cor. 3].

There exist several definitions of a *reduced* $\mathbb{Z}$-basis $(a_1, \dots, a_d)$ of $\mathbb{Z}_L$ in the literature, which are usually formulated in terms of the coefficients of the base-change matrix from $(a_1, \dots, a_d)$ to its Gram–Schmidt orthogonalization. We will use the following properties of such bases:

- If $(a_1, \dots, a_d)$ is *HKZ-reduced* [46, Chap. 2, Thm. 6], then for each $1 \leq k \leq d$, we have

$$\frac{4}{k+3} \leq \left(\frac{\|a_k\|}{\lambda_k}\right)^2 \leq \frac{k+3}{4}.$$

- If $(a_1, \dots, a_d)$ is *LLL-reduced* (with parameter $\delta = \frac{3}{4}$) [46, Chap. 2, Thm. 9], then for each $1 \leq k \leq d$, we have

$$\|a_k\| \leq 2^{(d-1)/2} \lambda_k.$$

  Moreover,

$$\prod_{k=1}^{d} \|a_k\| \leq 2^{d(d-1)/4} \Delta_L.$$

HKZ-reduced bases approximate the successive minima closely, but are difficult to compute as the dimension d grows. On the other hand, LLL-reduced bases can be computed in polynomial time in d [35].

Proposition 3.4. *There exist an algorithm and an absolute constant C such that the following holds. Let L be a number field of degree d and discriminant Δ_L. Let $(a_1, \dots, a_d)$ be an LLL-reduced basis of $\mathbb{Z}_L$, let $\mu_1, \dots, \mu_d$ be the complex embeddings of L, and let m_L be the matrix $(\mu_i(a_j))_{1 \leq i,j \leq d}$. Let $x \in \mathbb{Z}_L$, and let $H \geq 1$ such that $h(x) \leq H$. Let*

$$N \geq C(\log \Delta_L + dH + d^2).$$

Then, given approximations of $(\mu_i(x))_{1\le i\le d}$ *and* m_L^{-1} *to precision* N*, the algorithm computes* x *within* $O\big(d^2\mathcal{M}(H+\log\Delta_L+d^2)\big)$ *binary operations.*

Proof. Let $\lambda_j\in\mathbb{Z}$ such that $x=\sum\lambda_j a_j$. By definition of m_L, we have

$$\begin{pmatrix}\lambda_1\\ \vdots\\ \lambda_d\end{pmatrix}=m_L^{-1}\begin{pmatrix}\mu_1(x)\\ \vdots\\ \mu_d(x)\end{pmatrix}.$$

The determinant of m_L is Δ_L, so $|\det m_L|\ge 1$. In order to bound the absolute values of the coefficients of m_L^{-1} from above, we use Hadamard's lemma. Each coefficient of $(\det m_L)\cdot m_L^{-1}$ is the determinant of a submatrix of m_L, and the L^2-norms of the columns of m_L are precisely the $\|a_k\|$ for $1\le k\le d$. Moreover $\|a_k\|\ge 1$ for every k. Therefore,

$$|m_L^{-1}|\le\prod_{k=1}^{d}\|a_k\|\le 2^{d(d-1)/2}\Delta_L,$$

and hence

$$\log^+|m_L^{-1}|\le\log\Delta_L+O(d^2).$$

Since $h(x)\le H$, we have $\sum_{i=1}^d\log^+|\mu_i(x)|\le dH$. Therefore, for some choice of the constant C that we do not make explicit, we can recover the coefficients $\lambda_j\in\mathbb{Z}$ exactly. On average, we have $\log^+|\mu_i(x)|\le H$, so the cost of each multiplication is on average $O\big(\mathcal{M}(H+\log\Delta_L+d^2)\big)$ binary operations. Therefore the total cost of the matrix-vector product is only $O\big(d^2\mathcal{M}(H+\log\Delta_L+d^2)\big)$ binary operations. □

If $(a_1,\dots,a_d)$ is instead assumed to be HKZ-reduced in Proposition 3.4, then a similar proof shows that one can take

$$N\ge C(\log\Delta_L+dH+d\log d)$$

with a cost of $O\big(d^2\mathcal{M}(H+\log\Delta_L+d\log d)\big)$ binary operations. Indeed, in this case we have

$$\prod_{k=1}^{d}\|a_k\|\le d^d\prod_{k=1}^{d}\lambda_k\le d^d(1+\tfrac{d}{4})^{d/2}\Delta_L,$$

hence $\log^+|m_L^{-1}|\le\log(\Delta_L)+O(d\log d)$.

4 Computations in analytic moduli spaces

In this section, we describe the part of the evaluation algorithm for modular equations consisting in analytic computations in the Siegel and Hilbert moduli spaces. The different steps, in order of appearance, are:

1. The computation of a period matrix in Siegel space realizing a given triple of Igusa invariants;

2. In the Hilbert case, the computation of preimages under the Hilbert embedding (7);

3. The evaluation of theta constants at a given period matrix, which involves an approximate reduction to the Siegel fundamental domain.

In the first two steps, complexity estimates rely on the fact that the period matrices we encounter were initially obtained from algebraic data. The third step is purely analytic.

Before detailing these computations, we recall the classical definition of the Siegel fundamental domain $\mathcal{F}_2$ [32, Chap. I], and define certain open neighborhoods of this domain. Fix $\varepsilon \geq 0$, and let

$$Y = \begin{pmatrix} y_1 & y_3 \\ y_3 & y_2 \end{pmatrix}$$

be a symmetric 2×2 real matrix. Assume that Y is positive definite. We say that Y is *ε-Minkowski reduced* if

$$y_1 \leq (1+\varepsilon) y_2 \quad \text{and} \quad -\varepsilon y_1 \leq 2y_3 \leq (1+\varepsilon) y_1.$$

Let $\Sigma \subset \Gamma(1)$ be the set of 19 matrices defining the boundary of $\mathcal{F}_2$ described in [18]. We define the open set $\mathcal{F}_2^\varepsilon \subset \mathcal{H}_2$ as the set of all matrices $\tau \in \mathcal{H}_2$ such that

1. $\mathrm{Im}(\tau)$ is ε-Minkowski reduced,

2. $|\mathrm{Re}(\tau)| \leq 1/2 + \varepsilon$,

3. $|\det(\sigma^* \tau)| \geq 1 - \varepsilon$ for every $\sigma \in \Sigma$.

The fundamental domain $\mathcal{F}_2$ corresponds to the case $\varepsilon = 0$.

We also introduce the following notation. For $\tau \in \mathcal{H}_2$, we define

$$\Lambda(\tau) = \log \max\{2, |\tau|, \det(\operatorname{Im}\tau)^{-1}\}.$$

Denote by $\lambda_1(\tau) \leq \lambda_2(\tau)$ the two eigenvalues of $\operatorname{Im}(\tau)$, and by $m_1(\tau) \leq m_2(\tau)$ the successive minima of $\operatorname{Im}(\tau)$ on the lattice $\mathbb{Z}^2$. We also write

$$\Xi(\tau) = \log \max\{2, m_1(\tau)^{-1}, m_2(\tau)\}.$$

By [50, (5.4) p. 54], we always have

$$\frac{3}{4} m_1(\tau) m_2(\tau) \leq \det \operatorname{Im}(\tau) \leq m_1(\tau) m_2(\tau), \tag{11}$$

so that

$$\log \max\{\lambda_1(\tau)^{-1}, \lambda_2(\tau), m_1(\tau)^{-1}, m_2(\tau)\} = O(\Lambda(\tau)).$$

4.1 Computing period matrices

Let L be a number field of degree d_L, and fix a complex embedding μ of L. Given Igusa invariants $(j_1, j_2, j_3) \in L^3$, we wish to compute a period matrix $\tau \in \mathcal{F}_2$ realizing the Igusa invariants $\mu(j_1), \mu(j_2), \mu(j_3)$. We may assume that $j_3 \neq 0$: otherwise, modular equations are not defined at (j_1, j_2, j_3). Then τ can be computed in quasi-linear time using Borchardt means [14, §9.2.3]; our goal is to bound the precision losses in the process.

During the algorithm, we will consider a finite family $\Theta(j_1, j_2, j_3)$ of algebraic numbers constructed from j_1, j_2, j_3. More precisely we consider Θ as a finite family of polynomials $Q \in \mathbb{Q}[X_1, \ldots, X_n, Y]$, and the algebraic numbers we that consider are constructed as roots of polynomials of the form $Q(j_1, j_2, j_3, x_4, \ldots, x_n, Y)$ where $x_4, \ldots, x_n$ are previously constructed elements of $\Theta(j_1, j_2, j_3)$. When presented in this way, Θ does not depend on L, j_1, j_2, or j_3. As a toy example, consider the family Θ consisting of the single polynomial $X_1 - Y^2$; then $\Theta(j_1, j_2, j_3) = \{\sqrt{j_1}\}$. We call Θ a *finite recipe of algebraic extensions*. If H denotes the height of (j_1, j_2, j_3), then the height of all elements of $\Theta(j_1, j_2, j_3)$ is in $O_\Theta(H)$.

For every complex embedding μ of L, we define $B_{\Theta,\mu} \geq 0$ as the minimal real number such that

$$\left|\log(|\widetilde{\mu}(\theta)|)\right| \leq B_{\Theta,\mu}$$

for each nonzero $\theta \in \Theta(j_1, j_2, j_3)$ and each extension $\widetilde{\mu}$ of μ to the number field $L(\Theta(j_1, j_2, j_3))$. We can take $B_{\Theta,\mu} = O_\Theta(d_L H)$; moreover the sum of the bounds $B_{\Theta,\mu}$ over all the complex embeddings of L is also $O_\Theta(d_L H)$. A typical example of how we use $B_{\Theta,\mu}$ is the following.

Proposition 4.1. *There exist an algorithm and a finite recipe of algebraic extensions Θ such that the following holds. Let L be a number field, let $j_1, j_2, j_3 \in L$ be such that $j_3 \neq 0$, let μ be a complex embedding of L, and define $B_{\Theta,\mu}$ as above. Let $N \geq 1$. Then, given approximations of $\mu(j_k)$ for $1 \leq k \leq 3$ to precision N, the algorithm computes a genus 2 hyperelliptic curve $\mathcal{C}$ over $\mathbb{C}$ with Igusa invariants $\big(\mu(j_1), \mu(j_2), \mu(j_3)\big)$ for a cost of $O\big(\mathcal{M}(N + B_{\Theta,\mu})\big)$ binary operations, with a precision loss of $O(B_{\Theta,\mu})$ bits.*

Proof. Use Mestre's algorithm [38]. This algorithm involves $O(1)$ elementary operations with complex algebraic numbers constructed from the $\mu(j_k)$ for $1 \leq k \leq 3$, hence the estimates on the running time and precision loss. □

We first prove that the period matrix $\tau \in \mathcal{F}_2$ of $\mathcal{C}$ is bounded in terms of $B_{\Theta,\mu}$ for some acceptable choice of Θ. This is done by looking at theta quotients at τ. Recall from §2.1 that theta constants on $\mathcal{H}_2$ are denoted by θ_j for $0 \leq j \leq 15$.

Lemma 4.2. *There exists a finite recipe of algebraic extensions Θ such that the following holds. Let $\mathcal{C}$ be as in Proposition 4.1, and let $\tau \in \mathcal{F}_2$ be a period matrix of $\mathcal{C}$. Then we have*

$$|\tau| = O(B_{\Theta,\mu}).$$

Proof. By Thomae's formulas [44, Thm. IIIa.8.1], the quotients $\theta_j(\tau)/\theta_0(\tau)$ for $j \in [\![1, 15]\!]$ are algebraic numbers constructed from the coefficients of $\mathcal{C}$, and are nonzero for $j \in \{0, 1, 2, 3, 4, 6, 8, 12\}$. Therefore, we can choose Θ in such a way that

$$\big|\log\big(|\theta_j(\tau)/\theta_0(\tau)|\big)\big| \leq B_{\Theta,\mu} \quad \text{for } j \in \{4, 8\}.$$

Write $\tau = \begin{pmatrix} \tau_1 & \tau_3 \\ \tau_3 & \tau_2 \end{pmatrix}$. By [50, Prop. 7.7], we have

$$\begin{aligned} |\theta_0(\tau) - 1| &< 0.405, \\ |\theta_4(\tau)/\exp(i\pi\tau_1/4) - 1| &< 0.368, \\ |\theta_8(\tau)/\exp(i\pi\tau_2/4) - 1| &< 0.368. \end{aligned}$$

Therefore both $\mathrm{Im}(\tau_1)$ and $\mathrm{Im}(\tau_2)$ are in $O(B_{\Theta,\mu})$, hence also $|\mathrm{Im}(\tau_3)|$ because $\det \mathrm{Im}(\tau) > 0$. Since $\tau \in \mathcal{F}_2$, we have $|\mathrm{Re}(\tau)| \leq 1/2$ and the result follows. □

In order to compute τ from the values of theta quotients, we use Borchardt sequences [14, §9.2.3]. Define the matrices $J, M_1, M_2, M_3 \in \mathrm{Sp}_4(\mathbb{Z})$ whose actions on $\tau \in \mathcal{H}_2$ are given by

$$J\tau = -\tau^{-1}, \qquad M_1\tau = \tau + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$
$$M_2\tau = \tau + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad M_3\tau = \tau + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Let $\gamma_i = (JM_i)^2$ for $1 \leq i \leq 3$. Given theta quotients at τ, we compute the values of the modular functions

$$b_j = \theta_j^2/\theta_0^2, \quad 1 \leq j \leq 3$$

at τ and $\gamma_i\tau$ for $1 \leq i \leq 3$ using the transformation formula [43, §II.5]. We obtain the quantities

$$\frac{1}{\theta_0^2(\gamma_i\tau)}, \quad 1 \leq i \leq 3$$

as the limits of Borchardt sequences with *good sign choices* [30] starting from the tuples $\big(1, b_1(\gamma_i\tau), b_2(\gamma_i\tau), b_3(\gamma_i\tau)\big)$. Finally, we use that

$$\theta_0^2(\gamma_1\tau) = -i\tau_1\theta_4^2(\tau), \quad \theta_0^2(\gamma_2\tau) = -i\tau_2\theta_8^2(\tau), \quad \theta_0^2(\gamma_3\tau) = -\det(\tau)\theta_0^2(\tau). \tag{12}$$

In order to bound the complexity of this algorithm, we need estimates on the convergence of the Borchardt sequences above that are uniform in τ. We use the fact that theta constants converge quickly as the smallest eigenvalue $\lambda_1(\tau)$ of $\mathrm{Im}(\tau)$ tends to infinity. Moreover, $\lambda_1(\gamma_i\tau)$ is bounded from below when $\tau \in \mathcal{F}_2$ and $|\tau|$ is not too large.

Lemma 4.3. *For every $\tau \in \mathcal{H}_2$ such that $\lambda_1(\tau) \geq 1$, we have*

$$|\theta_j(\tau) - 1| < 4.18 \exp(-\pi\lambda_1(\tau)) \qquad \textit{for } 0 \leq j \leq 3.$$

Proof. Let $0 \leq j \leq 3$. Using the series expansion of θ_j, we obtain

$$|\theta_j(\tau) - 1| \leq \sum_{n \in \mathbb{Z}^2 \setminus \{0\}} \exp(-\pi n^t \,\mathrm{Im}(\tau) n) \leq \sum_{n \in \mathbb{Z}^2 \setminus \{0\}} \exp(-\pi\lambda_1(\tau)\|n\|^2).$$

By splitting the plane into quadrants, we see that this last sum is equal to $4S^2 + 4S$, with

$$S = \sum_{n \geq 1} \exp(-\pi \lambda_1(\tau) n^2) \leq \frac{\exp(-\pi \lambda_1(\tau))}{1 - \exp(-3\pi \lambda_1(\tau))}.$$

Since $\lambda_1(\tau) \geq 1$, the conclusion follows. □

Lemma 4.4. *Let $\tau \in \mathcal{H}_2$ and $\gamma \in \mathrm{Sp}_4(\mathbb{Z})$. Then*

$$\lambda_1(\gamma\tau) \geq \frac{\det \mathrm{Im}(\tau)}{8|\gamma|^2 |\tau| (2|\tau| + 1)^2}.$$

Proof. We have

$$\lambda_1(\gamma\tau) \geq \frac{\det \mathrm{Im}(\gamma\tau)}{\mathrm{Tr}\, \mathrm{Im}(\gamma\tau)}.$$

By [50, (5.11) p. 57], we also have

$$\mathrm{Im}(\gamma\tau) = (\gamma^*\tau)^{-t} \mathrm{Im}(\tau) (\gamma^* \bar{\tau})^{-1}$$

so that

$$\det \mathrm{Im}(\gamma\tau) = \frac{\det \mathrm{Im}(\tau)}{|\det(\gamma^*\tau)|^2}, \quad \text{and}$$

$$\mathrm{Tr}\, \mathrm{Im}(\gamma\tau) \leq 8 |(\gamma^*\tau)^{-1}|^2 |\mathrm{Im}(\tau)| \leq 8 \frac{|\gamma^*\tau|^2 |\tau|}{|\det(\gamma^*\tau)|^2} \leq 8 \frac{|\gamma|^2 (2|\tau| + 1)^2 |\tau|}{|\det(\gamma^*\tau)|^2}.$$ □

Proposition 4.5. *There is an algorithm such that the following holds. Let $\tau \in \mathcal{F}_2$ and $N \geq 1$. Then, given approximations of squares of theta quotients at τ to precision N, the algorithm computes an approximation of τ within*

$$O\big(\mathcal{M}(N + |\tau|) \log |\tau| + \mathcal{M}(N) \log N\big)$$

binary operations. The precision loss is $O(\log N + |\tau| \log |\tau|)$ bits.

Proof. We obtain the quantities

$$(\theta_j^2(2^n \gamma_i \tau) / \theta_0^2(\gamma_i \tau))_{0 \leq j \leq 3}$$

after n Borchardt steps. By Lemma 4.4, we know that

$$|\log \lambda_1(\gamma_i \tau)| = O(\log |\tau|).$$

Therefore, we can choose $n = O(\log|\tau|)$ such that $\lambda_1(2^n\gamma_i\tau) \geq 10$, for instance. Up to this point, we performed $O(\log|\tau|)$ elementary operations with complex numbers z such that $\log|z| = O(|\tau|)$. Therefore the total cost is

$$O(\mathcal{M}(N + |\tau|)\log|\tau|)$$

binary operations, and the precision loss is $O(|\tau|\log|\tau|)$ bits. Even if $|\tau|$ is not known explicitly, this moment is detected in the algorithm as when the four values in the Borchardt sequence get very close to each other.

Then, we normalize so that one of the four values is 1, and continue performing a further $O(\log N)$ Borchardt steps: this O-constant and the accuracy of the result can be made explicit by [14, Prop. 7.2]. This costs $O(\mathcal{M}(N)\log N)$ binary operations, and the precision loss is $O(\log N)$ bits. This allows us to compute the quantities $\theta_0^2(\gamma_i\tau)$ for $1 \leq i \leq 3$; the precision loss up to now is $O(\log N + |\tau|\log|\tau|)$ bits. Finally, we recover the entries of τ using eq. (12). This final computation costs $O(N + |\tau|)$ binary operations, and the precision loss is $O(|\tau|)$ bits. □

Theorem 4.6. *There exists an algorithm and a finite recipe of algebraic extensions Θ such that the following holds. Let L be a number field, let $j_1, j_2, j_3 \in L$ be such that $j_3 \neq 0$, let μ be a complex embedding of L, and define $B_{\Theta,\mu}$ as above. Let $N \geq 1$. Then, given approximations of $\mu(j_k)$ for $1 \leq k \leq 3$ to precision N, the algorithm computes a matrix $\tau \in \mathcal{F}_2$ such that the Igusa invariants at τ are the $\mu(j_k)$ for $1 \leq k \leq 3$. The algorithm involves $O\big(\mathcal{M}(N + B_{\Theta,\mu})\log(N + B_{\Theta,\mu})\big)$ binary operations, and a precision loss of $O(\log N + B_{\Theta,\mu}\log B_{\Theta,\mu})$ bits.*

Proof. First, we compute a curve $\mathcal{C}$ as in Proposition 4.1. Then, by Thomae's formula, there is a finite number of possibilities for the values of squares of theta quotients at τ; one of them corresponds to an actual $\tau \in \mathcal{F}_2$, and the others correspond to other elements in the orbit $\Gamma(1)\tau$. When we attempt to run the algorithm of Proposition 4.5 on these inputs, we may discard all resulting period matrices that do not belong to $\mathcal{F}_2$. In order to distinguish between the remaining possible values of τ, it is usually enough to compute theta constants to precision $O(1)$ using the naive algorithm, and match with the input. In extreme cases, we may resort to computing Igusa invariants at all remaining possible values of τ to precision $O(N + B_\mu)$ for the cost of $O\big(\mathcal{M}(N + B_\mu)\log(N + B_\mu)\big)$ binary operations [27, Thm. 5.2] □

4.2 Inverting the Hilbert embedding

Let F be a real quadratic field, and let $R \in \mathrm{GL}_2(\mathbb{R})$ be the matrix defining the Hilbert embedding Φ_F as in 2.2. Recall that for every $\tau \in \mathcal{H}_2$, we denote by $0 < \lambda_1(\tau) \leq \lambda_2(\tau)$ the two eigenvalues of $\mathrm{Im}(\tau)$.

Lemma 4.7. *There exists a constant $C > 0$ depending on F such that for every $t = (t_1, t_2) \in \mathcal{H}_1^2$, we have*

$$\frac{1}{C}\lambda_1(\Phi_F(t)) \leq \min\{\mathrm{Im}(t_1), \mathrm{Im}(t_2)\} \leq C\lambda_1(\Phi_F(t)),$$
$$\frac{1}{C}\lambda_2(\Phi_F(t)) \leq \max\{\mathrm{Im}(t_1), \mathrm{Im}(t_2)\} \leq C\lambda_2(\Phi_F(t)).$$

Proof. This is obvious from the formula (7). □

Theorem 4.8. *Let F be a real quadratic field, and let R be as above. Then there exist an algorithm, a constant $C > 0$, and a finite recipe of algebraic extensions Θ depending on F such that the following holds. Let L be a number field, let $j_1, j_2, j_3 \in L$ be such that $j_3 \neq 0$, let μ be a complex embedding of L, and define $B_{\Theta,\mu}$ as in §4.1. Let $\mathcal{C}$ be a genus 2 hyperelliptic curve over $\mathbb{C}$ with Igusa invariants $\mu(j_1), \mu(j_2), \mu(j_3)$, and assume that $\mathrm{Jac}(\mathcal{C})$ has real multiplication by $\mathbb{Z}_F$. Let $\tau \in \mathcal{F}_2$ be a period matrix of $\mathcal{C}$. Then there exists $t = (t_1, t_2) \in \mathcal{H}_1^2$ such that $\Phi_F(t) \in \mathcal{H}_2$ is a period matrix of $\mathcal{C}$, and*

$$|\log(\mathrm{Im}\, t_i)| \leq C B_{\Theta,\mu} \quad \text{for } i = 1, 2.$$

Moreover, given an approximation of τ to precision $N + CB_{\Theta,\mu}$, the algorithm computes t to precision N within $\widetilde{O}_F(N + B_{\Theta,\mu})$ binary operations.

Proof. By Lemma 4.2, if Θ is well chosen, we have

$$|\tau| = O(B_{\Theta,\mu}).$$

The result would follow directly from Lemma 4.7 if there existed $t \in \mathcal{H}_1^2$ such that $\tau = \Phi_F(t)$, but this is not always the case. In general, by [2, Lem. 4.1], there exist coprime integers a, b, c, d, e such that

$$b^2 - 4ac - 4de = \Delta_F \quad \text{and} \quad a\tau_1 + b\tau_3 + c\tau_2 + d\det(\tau) + e = 0. \tag{13}$$

We claim that the heights of a, b, c, d, e must be in $O_F(B_{\Theta,\mu})$ for some choice of Θ. We prove this by comparing the analytic and rational representations (see [1, §1.2]) of the endomorphism $\sqrt{\Delta_F}$ on the complex abelian variety

$$A_\tau = \mathbb{C}^2/(\tau\mathbb{Z}^2 \oplus \mathbb{Z}^2).$$

By [2, Cor. 4.2], the rational representation of an endomorphism f in the image of $\mathbb{Z}_F$ inside $\mathrm{End}(A_\tau)$ is of the form

$$\rho_{R,\tau}(f) = \begin{pmatrix} n & ma & 0 & md \\ -mc & mb+n & -md & 0 \\ 0 & me & n & -mc \\ -me & 0 & ma & mb+n \end{pmatrix} \qquad \text{for some } m, n \in \mathbb{Z}.$$

On the other hand, the analytic representation $\rho_{A,\tau}(\sqrt{\Delta_F})$ of the endomorphism $\sqrt{\Delta_F}$ of A_τ can be computed as follows. Let $\omega = (\omega_1, \omega_2)$ be a basis of differential forms on A_τ such that $\mathrm{Sym}^2(\omega)$ corresponds by the Kodaira–Spencer isomorphism to a deformation of A_τ along the Humbert surface. Then, the analytic representation of $\sqrt{\Delta_F}$ in the basis ω is of the form

$$\pm \begin{pmatrix} \sqrt{\Delta_F} & 0 \\ 0 & -\sqrt{\Delta_F} \end{pmatrix}.$$

Algorithm 4.6 in [31] shows that such a basis ω exists; moreover the base change matrix m between (dz_1, dz_2) and ω can be chosen such that

$$\log \max\{|m|, |m^{-1}|\} = O_F(B_{\Theta,\mu}).$$

after extending Θ in a suitable way. Therefore, the analytic representation of $\sqrt{\Delta_F}$ on A_τ satisfies

$$\log^+ \left|\rho_{A,\tau}(\sqrt{\Delta})\right| = O_F(B_{\Theta,\mu}).$$

For every $f \in \mathrm{End}(A_\tau)$, the rational and analytic representations of f are related by the following formula [1, Rem. 8.14]:

$$\rho_{A,\tau}(f)(\tau\ I_2) = (\tau\ I_2)\rho_{R,\tau}(f).$$

Taking imaginary parts, we find that there exist $m, n \in \mathbb{Z}$ such that

$$\mathrm{Im}(\tau) \begin{pmatrix} n & ma \\ -mc & mb+n \end{pmatrix} = \mathrm{Im}\big(\rho_{A,\tau}(\sqrt{\Delta_F})\tau\big),$$
$$\mathrm{Im}(\tau) \begin{pmatrix} 0 & md \\ -md & 0 \end{pmatrix} = \mathrm{Im}\big(\rho_{A,\tau}(\sqrt{\Delta_F})\big).$$

Therefore the heights of a, b, c, d, m, n are in $O_F(B_{\Theta,\mu})$. The same is true for e by (13). This proves our claim.

The algorithm to compute t is the following. We compute the integers a, b, c, d, e in $\widetilde{O}_F(B_{\Theta,\mu})$ binary operations with the LLL algorithm [47], using eq. (13). Then, we use the algorithm from [2, Prop. 4.5] to compute a matrix $\gamma \in \Gamma(1)$ such that $\gamma\tau$ lies in the image of Φ_F; the matrix γ has a simple expression in terms of a, b, c, d, e, hence we also have

$$\log|\gamma| = O_F(B_{\Theta,\mu}).$$

By Lemma 4.4, we also have

$$\Lambda(\gamma\tau) \in O_F(B_{\Theta,\mu}),$$

so the result follows from Lemma 4.7. □

4.3 Computing theta constants

We now turn to the problem of evaluating theta constants at a given period matrix. We crucially rely on the following fact: given $\tau \in \mathcal{F}_2$ (for instance a matrix with dyadic entries), one can evaluate theta constants at τ to any desired precision $N \geq 0$ in uniform quasi-linear time $O(\mathcal{M}(N)\log N)$ [27, Thm. 5.2]. When given a general $\tau \in \mathcal{H}_2$, the standard strategy is to *reduce* τ and compute a symplectic matrix $\gamma \in \Gamma(1)$ such that $\gamma\tau$ is very close to $\mathcal{F}_2$. Then we increase the imaginary parts of the coefficients of $\gamma\tau$ slightly to obtain an actual period matrix $\tau' \in \mathcal{F}_2$. Computing theta constants at τ' yields good approximations of the value of theta constants at $\gamma\tau$.

We now describe the approximate reduction algorithm, which mimics [51, Alg. 6.8]. The input consists a matrix $\tau \in \mathcal{H}_2$ to some precision $N \geq 1$, and the output is a matrix $\gamma \in \mathrm{Sp}_4(\mathbb{Z})$ such that $\gamma\tau \in \mathcal{F}_2^\varepsilon$. We assume that the current precision remains greater than $|\log_2 \varepsilon| + 1$ at any time; if we run out of precision, we stop and output "failure".

Algorithm 4.9 (Reduction to $\mathcal{F}_2^\varepsilon$)**.** Start with $\tau' = \tau$ and iterate the following three steps until $\tau' \in \mathcal{F}_2^\varepsilon$, keeping track of a matrix $\gamma \in \Gamma(1)$ such that $\tau' = \gamma\tau$:

1. Reduce $\mathrm{Im}(\tau')$ such that it becomes ε-Minkowski reduced.
2. Reduce $\mathrm{Re}(\tau')$ such that $|\mathrm{Re}(\tau')| \leq 1/2 + \varepsilon$.
3. Apply $\sigma \in \Sigma$ such that $|\det \sigma^*(\tau')|$ is at most $1 - \varepsilon/2$ and minimal, if such a σ exists.

4. Update $\gamma \in \Gamma(1)$ and recompute $\tau' = \gamma\tau$.

In order to analyze Algorithm 4.9, we mimic the analysis of the exact reduction algorithm [51, §6]. First, we detail the Minkowski reduction step.

Lemma 4.10. *There exists an algorithm and an absolute constant C such that the following holds. Let $\tau \in \mathcal{H}_2$ and $\varepsilon > 0$. Then, given an approximation of τ to precision $N \geq C\Lambda(\tau) + |\log_2 \varepsilon|$, the algorithm computes a matrix $U \in \mathrm{SL}_2(\mathbb{Z})$ such that $U^t \operatorname{Im}(\tau) U$ is ε-Minkowski reduced within $O(\mathcal{M}(N) \log N)$ binary operations.*

Proof. Write $\operatorname{Im}(\tau) = R^t R$, and consider the matrix R' obtained by rounding the coefficients of $2^N R$ to the nearest integers. If C is chosen appropriately, then the matrix R' is still invertible. We apply a quasi-linear version of Gauss's reduction algorithm to R' [58], and obtain a reduced basis of the lattice $R'\mathbb{Z}^2$ within $O(\mathcal{M}(N) \log N)$ binary operations. The base change matrix $U \in \mathrm{SL}_2(\mathbb{Z})$ must satisfy

$$\log|U| = O(\Lambda(\tau))$$

by [51, Lem. 6.6]. Therefore, the matrix $U^t \operatorname{Im}(\tau) U$ will be ε-Minkowski reduced if C is large enough. □

Then, we bound precision losses during the execution of Algorithm 4.9.

Lemma 4.11. *Let $\tau, \tau' \in \mathcal{H}_2$, and assume that there exists $\gamma \in \Gamma(1)$ such that $\tau' = \gamma\tau$. Then we have*

$$\log^+ \max\{|\gamma^*\tau|, |(\gamma^*\tau)^{-1}|\} = O(\max\{\Lambda(\tau), \Lambda(\tau')\}),$$
$$\log|\gamma| = O(\max\{\Lambda(\tau), \Lambda(\tau')\}).$$

Proof. Let R be a real matrix such that $R^t R = \operatorname{Im}(\tau)$. Then we have

$$\operatorname{Im}(\tau') = (\gamma^*\tau)^{-t} \operatorname{Im}(\tau) (\gamma^*\overline{\tau})^{-1} = R'^t \overline{R'}$$

with $R' = R(\gamma^*\tau)^{-1}$. Since $|R| \leq |\operatorname{Im}(\tau)|^{1/2}$ and $|R'| \leq |\operatorname{Im}(\tau')|^{1/2}$, we obtain

$$|\gamma^*\tau| = |R'^{-1}R| \leq 2\frac{|R'|}{\det(R')}|R|$$

so $\log^+|\gamma^*\tau| = O(\max\{\Lambda(\tau), \Lambda(\tau')\})$, and similarly for $(\gamma^*\tau)^{-1}$.

It remains to bound $|\gamma|$. If c, d denote the two lower blocks of γ, then we have $\operatorname{Im}(\gamma^*\tau) = c \operatorname{Im}(\tau)$. Therefore $\log^+|c| = O(\max\{\Lambda(\tau), \Lambda(\tau')\})$, and in turn $\log^+|d| \leq \log^+(|c\tau| + |\gamma^*\tau|) = O(\max\{\Lambda(\tau), \Lambda(\tau')\})$. Finally, we bound the upper blocks a and b of γ using the relation $a\tau + b = \tau'(c\tau + d)$. □

Lemma 4.12. *There is an absolute constant C such that the following holds. Let $\tau \in \mathcal{H}_2$ and $\varepsilon > 0$, and assume that the precision during Algorithm 4.9 remains greater than $|\log_2 \varepsilon|+1$. Then the number of loops is $O(\Xi(\tau))$. Moreover, during the algorithm, the quantities $\big|\log\big(|\det(\gamma^*\tau)|\big)\big|$, $\Lambda(\tau')$ and $\log|\gamma|$ remain in $O(\Lambda(\tau))$.*

Proof. The number of iterations is $O(\Xi(\tau))$ by [51, Prop. 6.16]: observe that [51, Lem. 6.11 and 6.12] still apply, because $\det \operatorname{Im}(\tau')$ is strictly increasing in Algorithm 4.9. The proof of [51, Lem. 6.16] also applies to Algorithm 4.9 with slightly worse constants. This shows that $\log|\tau'|$ and $\log|\det(\gamma^*\tau)|$ remain in $O(\Lambda(\tau))$.

During the algorithm, we have $\log^+ m_2(\tau') = O(\Lambda(\tau))$ by [51, Lem. 6.13]. Moreover $\det \operatorname{Im}(\tau') \geq \det \operatorname{Im}(\tau)$, so

$$m_1(\tau')^{-1} \leq \frac{m_2(\tau')}{\det \operatorname{Im}(\tau')} \leq \frac{m_2(\tau')}{\det \operatorname{Im}(\tau)} \leq \frac{4m_2(\tau')}{3m_1(\tau)^2}$$

by (11). Therefore we also have $\Lambda(\tau') = O(\Lambda(\tau))$. The remaining bounds follow from Lemma 4.11. □

Proposition 4.13. *There is an absolute constant C such that the following holds. Let $\tau \in \mathcal{H}_2$ and $\varepsilon > 0$. Then, given an approximation of τ to precision $N \geq C\Lambda(\tau) + |\log_2 \varepsilon|$ as input, Algorithm 4.9 does not run out of precision, and computes a matrix $\gamma \in \Gamma(1)$ such that $\gamma\tau \in \mathcal{F}_2^\varepsilon$ and $\log|\gamma| = O(\Lambda(\tau))$. It costs $O(\Xi(\tau)\mathcal{M}(N)\log N)$ binary operations.*

Proof. By Lemma 4.12, there is a constant C' such that $\log|\gamma| \leq C'\Lambda(\tau)$ during the execution of Algorithm 4.9 as long as the absolute precision is at least $|\log_2 \varepsilon| + 1$. Therefore, if C is chosen appropriately, step 4 in Algorithm 4.9 ensures that the absolute precision is at least $|\log_2 \varepsilon| + 1$ at every step. Hence the estimate on $\log|\gamma|$ and $\Lambda(\tau')$ remains valid until the end of the algorithm, and we can perform the approximate Minkowski reductions using Lemma 4.10. By Lemma 4.12, there are $O(\Xi(\tau))$ steps in Algorithm 4.9, and by Lemma 4.10, each step costs $O(\mathcal{M}(N)\log N)$ binary operations. When the algorithm stops, the absolute precision is still greater than $|\log_2 \varepsilon| + 1$, so the final τ' belongs to $\mathcal{F}_2^\varepsilon$. □

Theorem 4.14. *There exist an algorithm and an absolute constant C such that the following holds. Let $\tau \in \mathcal{H}_2$ and $N \geq 1$. Then, given an approximation of τ to precision $N + C\Lambda(\tau)$, the algorithm computes*

1. *a matrix* $\gamma \in \mathrm{Sp}_4(\mathbb{Z})$ *such that* $\log|\gamma| = O(\Lambda(\tau))$,

2. *a matrix* $\tau' \in \mathcal{F}_2$ *such that* τ' *is an approximation of* $\gamma\tau$ *to precision* N,

3. *an approximation of squares of theta constants at* $\gamma\tau$ *to precision* N,

within

$$O\big(\Xi(\tau)\mathcal{M}(\Lambda(\tau))\log\Lambda(\tau) + \mathcal{M}(N)\log N\big)$$

binary operations.

Proof. Fix $\varepsilon = 10^{-2}$, for instance. First, we apply Proposition 4.13 to compute γ such that $\gamma\tau \in \mathcal{F}_2^{\varepsilon}$, using $O\big(\Xi(\tau)\mathcal{M}(\Lambda(\tau))\log\Lambda(\tau)\big)$ binary operations. Then, we recompute $\gamma\tau$ to high precision, and reduce it further if necessary (using $O(1)$ loops in the reduction algorithm 4.9) to land in $\mathcal{F}_2^{\varepsilon'}$ where $\varepsilon' = 2^{-N}\exp(C'\Lambda(\tau))$ for some appropriate constant C'. This further reduction step costs $O(\mathcal{M}(N + \Lambda(\tau)))$ binary operations. After that, we can increase the imaginary parts of the coefficients of $\gamma\tau$ slightly and obtain $\tau' \in \mathcal{F}_2'$ such that

$$|\tau' - \gamma\tau| \leq C''\varepsilon'|\gamma\tau|$$

for some absolute constant C''. We output squares of theta constants evaluated at τ' to precision $N+1$ using [27, Thm. 5.2]. Since derivatives of theta constants are uniformly bounded on a neighborhood on $\mathcal{F}_2$, the result is a suitable approximation of squared theta constants at $\gamma\tau$. □

5 Evaluating Hilbert modular equations

In this final section, we describe the complete algorithm to evaluate Hilbert modular equations in Gundlach invariants for $F = \mathbb{Q}(\sqrt{5})$. The algorithm is easily adapted to handle different choices of quadratic fields and invariants. The case of Siegel modular equations is even simpler, since the Hilbert embedding does not appear; we do not detail it and simply point out the differences in running time.

5.1 Analytic evaluation of modular equations

Let L be a number field, let $(g_1, g_2) \in L$ be a pair of Gundlach invariants, and let (j_1, j_2, j_3) be the associated Igusa invariants. We may assume that $g_1 \neq 0$ (i.e. $j_3 \neq 0$); otherwise, the denominator of modular equations

vanishes. Choose β and ℓ as in §2.2, and let μ be a complex embedding of L. The following algorithm computes the numerator and denominator of Hilbert modular equations of level β in Gundlach invariants evaluated at $(\mu(g_1), \mu(g_2))$.

Algorithm 5.1. 1. Compute a period matrix $\tau \in \mathcal{F}_2$ with Igusa invariants $(\mu(j_1), \mu(j_2), \mu(j_3))$ using Theorem 4.6.

2. Compute a matrix $\gamma \in \mathrm{Sp}_4(\mathbb{Z})$ and $t \in \mathcal{H}_1^2$ such that $\Phi_F(t) = \gamma\tau$ using Theorem 4.8.

3. Consider the following set of symplectic matrices:

$$C_\tau = \Phi_F\left(\left(\begin{smallmatrix} 1 & 0 \\ 0 & \beta \end{smallmatrix}\right)\eta\right) \cdot \gamma \in \mathrm{GSp}_4(\mathbb{Q}).$$

   Apply Proposition 2.3 to transform C_τ into another set of representatives $D_\tau \subset \mathrm{GSp}_4(\mathbb{Q})$ for the same orbits under $\Gamma(1)$, consisting of matrices whose lower-left block is zero.

4. Compute squares of theta constants at all period matrices of the form $\eta\tau$ for $\eta \in D_\tau$ using Theorem 4.14.

5. Build product trees as described in Lemma 3.1 to compute the numerators and denominator of Hilbert modular equations evaluated at t, using the formulas of §2.2.

To avoid complicated expressions, we hide logarithmic factors in the $\widetilde{O}$ notation from now on. Recall that $Q_\beta \in \mathbb{Q}[J_1, J_2]$ denotes the denominator of Hilbert modular equations $\Psi_{\beta,k}$ for $1 \le k \le 2$ constructed in §2.2.

Proposition 5.2. *There exists a finite recipe of algebraic extensions Θ and a constant $C > 0$ such that the following holds. Let L be a number field, let μ be a complex embedding of L, let $(g_1, g_2) \in L^2$ be such that $g_1 \neq 0$, define $B_{\Theta,\mu}$ as in §4.1, and let $N \ge C(B_{\Theta,\mu} \log B_{\Theta,\mu} + \log \ell)$. Then, given approximations of $\mu(g_1)$ and $\mu(g_2)$ to precision N, Algorithm 5.1 computes $\mu(Q_\beta(g_1, g_2))$ and $\mu(Q_\beta \Psi_{\beta,k}(g_1, g_2))$ for $k \in \{1, 2\}$ within $\widetilde{O}(\ell B_{\Theta,\mu} + \ell N)$ binary operations, with a precision loss of $\widetilde{O}(\ell B_{\Theta,\mu} + \log N)$ bits.*

Proof. Steps 1 and 2 can be performed in $\widetilde{O}(N + B_{\Theta,\mu})$ binary operations with a precision loss of $O(\log N + B_{\Theta,\mu} \log B_{\Theta,\mu})$ bits, by the results cited above. We have $\log|\gamma| = O(B_{\Theta,\mu})$; therefore the elements of C_τ have coefficients

of height $O(B_{\Theta,\mu} + \log \ell)$. In step 3, the set D_τ can then be computed in $\widetilde{O}(\ell B_{\Theta,\mu})$ binary operations by Proposition 2.3. For each $\eta \in D_\tau$, we have

$$\Xi(\eta\tau) = O(\log \ell).$$

By Theorem 4.14, the precision loss taken in the reduction algorithm applied on each $\eta\tau$ is $O(B_{\Theta,\mu} + \log \ell)$; the total cost of reduction is $\widetilde{O}(\ell(N + B_{\Theta,\mu}))$ binary operations. Let $\gamma_\eta \in \Gamma(1)$ be the reduction matrix provided by the theorem. The total cost of computing all squared theta constants at the period matrices $\tau_\eta = \gamma_\eta(\eta\tau)$ in step 4 is $\widetilde{O}(\ell N)$. This yields the values of the modular forms h_4, h_6, h_{10}, h_{12} at the matrices τ_η using $O(\ell)$ binary operations, with a further precision loss of $O(1)$ bits.

In step 5, we evaluate $g_\beta(t)$ using eq. (10), and the relations satisfied by h_{10} as a modular form: for instance, we have

$$h_{10}^2(\eta\tau) = (\det \gamma_\eta^*(\eta\tau))^{-20} h_{10}^2(\tau_\eta),$$

for each $\eta \in D_\tau$. By Lemma 4.11, the total precision loss in this computation is $O(\ell(B_{\Theta,\mu} + \log \ell))$; the total cost of computing $g_\beta(t)$ is $\widetilde{O}(\ell(N + B_{\Theta,\mu}))$ binary operations. Up to a similar scalar factor, the polynomials $\mu(\Psi_{\beta,k}(g_1, g_2))$ for $k \in \{1, 2\}$ are given by

$$\prod_{\eta \in C_\beta^\sigma} (F_{10}(\tau_\eta)X + G_2^5(\tau_\eta)) \qquad \text{for } k = 1,$$

$$\sum_{\eta \in C_\beta^\sigma} G_2^5(\tau_\eta) \prod_{\eta' \in C_\beta^\sigma \setminus \{\eta\}} (F_{10}(\tau_{\eta'})X - G_2^2(\tau_{\eta'})F_6(\tau_{\eta'})) \qquad \text{for } k = 2.$$

By Lemma 3.1, these polynomials can be computed in $\widetilde{O}(\ell N)$ binary operations, with a precision loss of $O(\ell)$ bits. We conclude by summing precision losses and binary costs of each step. □

In the case of Siegel modular equations, the complexity and precision loss estimates are similar, with each occurence of ℓ replaced by ℓ^3.

5.2 Algebraic reconstruction

Once modular equations and their denominators have been computed in every complex embedding, we only have to recognize their coefficients as algebraic numbers. We present two results, one in the case of a finite field (generalizing Theorem 1.2), and the second in the case of a number field.

In the case of a finite field, we are given a prime power $q = p^d$, and a monic polynomial $P \in \mathbb{Z}[X]$ of degree d, irreducible modulo p. We let $M \geq 1$ such that $\log|P| \leq M$. We assume that a black box provides us with approximations of the roots of P to any desired precision. Then, we represent elements of $\mathbb{F}_q$ as elements of $\mathbb{F}_p[X]/(P)$.

Theorem 5.3. *There exists an algorithm such that the following holds. Let $\beta \in \mathbb{Z}_F$ for $F = \mathbb{Q}(\sqrt{5})$ or $\mathbb{Q}(\sqrt{8})$ be a totally positive prime, prime to the discriminant Δ_F, of prime norm $\ell \in \mathbb{Z}$. Then, given ℓ and $g_1, g_2 \in \mathbb{F}_q$ where the denominator of Hilbert modular equations $\Psi_{\beta,k}$ does not vanish, the algorithm computes $\Psi_{\beta,k}(g_1, g_2) \in \mathbb{F}_q[X]$ for $k \in \{1, 2\}$ within*

$$\widetilde{O}(\ell^2 d \log p + \ell^2 d^2 M)$$

binary operations.

If $dM = O(\log p)$, and in particular when $q = p$ is prime, the cost estimate simplifies to $\widetilde{O}(\ell^2 \log q)$ binary operations. Theorem 1.1 stated in the introduction is the analogue of Theorem 5.3 for Siegel modular equations over prime finite fields, and is obtained by a similar proof, except that each occurrence of ℓ should be replaced by ℓ^3.

Proof. Let L be the number field $\mathbb{Q}[X]/(P)$, and let α be a root of P in L. We lift g_1 and g_2 to elements of $\mathbb{Z}[\alpha]$ in such a way that the height of their coefficients is bounded above by $\log p$. Then the following inequalities hold:

$$\begin{aligned} h(\alpha) &\leq M + \log 2, \quad \text{and} \\ \max\{h(g_1), h(g_2)\} &\leq \log(p) + dh(\alpha) + \log(d) = O(dM + \log p). \end{aligned}$$

Since Q_β and the coefficients of $Q_\beta \Psi_{\beta,k}$ are polynomials in $\mathbb{Z}[g_1, g_2]$ of degree $O(\ell)$ and height $O(\ell \log \ell)$, the algebraic numbers we have to recognize are all elements of $\mathbb{Z}[\alpha]$, and the height of their coefficients is $O(\ell \log \ell + \ell d M + \ell \log p)$. By Proposition 3.3, we can recognize each coefficient within $\widetilde{O}(\ell d^2 M + \ell d \log p)$ binary operations, provided that its images under all complex embeddings of L are computed to precision $C(\ell \log \ell + \ell d M + \ell \log p)$, where C is a suitable absolute constant.

Let μ be a complex embedding of L, and choose a starting precision N. Then $\mu(g_1)$ and $\mu(g_2)$ are obtained by replacing α by one of the complex roots of P: this can be done within $\widetilde{O}(d(M + N))$ binary operations, and

a precision loss of $O(dM + \log p)$ bits, using Horner's algorithm. We run Algorithm 5.1 for each complex embedding μ of L; let Θ be the finite recipe of algebraic extensions provided by Proposition 5.2. It suffices to choose N in

$$\widetilde{O}(\ell dM + \ell \log p + \ell B_{\Theta,\mu}).$$

The cost of Algorithm 5.1 is then $\widetilde{O}(\ell^2 dM + \ell^2 \log p + \ell B_{\Theta,\mu})$ binary operations, for each μ. Since we have

$$\sum_{\mu} B_{\Theta,\mu} = O(d \log p + d^2 M),$$

total cost of analytic evaluations over all embeddings is $\widetilde{O}(\ell^2 d \log p + \ell^2 d^2 M)$ binary operations, and dominates the cost of algebraic reconstruction. □

If $g_1, g_2 \in \mathbb{Z}$ are small integers, then the complexity of evaluating modular equations is quasi-linear in the output size.

Theorem 5.4. *There exists an algorithm such that the following holds. Given the prime ℓ and $g_1, g_2 \in \mathbb{Z}$ such that*

$$\max\{|g_1|, |g_2|\} \in O(1) \quad \textit{and} \quad Q_\beta(g_1, g_2) \neq 0,$$

the algorithm computes the polynomials $\Psi_{\beta,k}(g_1, g_2) \in \mathbb{Q}[X]$ for $k \in \{1, 2\}$ within $\widetilde{O}(\ell^2)$ binary operations.

Proof. In this case, we have $B_{\Theta,\mu} = O(1)$. It is sufficient to round the result of Proposition 5.2 with $N = C\ell \log \ell$, where C is an absolute constant, to the nearest integers. □

The complexity of evaluating Hilbert modular equations over a number field of degree d over $\mathbb{Q}$ can also be bounded above in terms of the discriminant and the height of the operands. We assume that an LLL-reduced integer basis of the number field L has been precomputed. Moreover, if m_L is the matrix defined in Proposition 3.4, we assume that a black box provides us with the coefficients of m_L^{-1} to any desired precision.

Theorem 5.5. *There exists an algorithm such that the following holds. Let $H \geq 1$, and let $g_1, g_2 \in L$ given as quotients of integers of height at most H*

such that $Q_\beta(g_1, g_2) \neq 0$. *Then the algorithm computes* $\Psi_{\beta,k}(g_1, g_2) \in L[X]$ *for* $k \in \{1, 2\}$ *within*

$$\widetilde{O}(\ell^2 d^2 H + \ell d^2 \log \Delta_L + \ell d^4)$$

binary operations.

In the case $L = \mathbb{Q}$, the cost estimate simplifies to $\widetilde{O}(\ell^2 H)$ binary operations, thus generalizing Theorem 5.4.

Proof. For simplicity, assume that g_1 and g_2 are actually integers: in the general case we multiply Q_β by an appropriate power of a common denominator of g_1 and g_2 in $\mathbb{Z}_L$. We know that Q_β and the coefficients of $Q_\beta \Psi_{\beta,k}$ are polynomials in $\mathbb{Z}[J_1, J_2]$ of degree $O(\ell)$ and height $O(\ell \log \ell)$: their evaluations at (g_1, g_2) are therefore algebraic integers of height $\widetilde{O}(\ell H)$. By Proposition 3.4, we can recognize each coefficient within $\widetilde{O}(d^2 \ell H + d^2 \log \Delta_L + d^4)$ binary operations, provided that complex approximations are computed to a high enough precision N. It suffices to take N in $\widetilde{O}(\log \Delta_L + d\ell H + d^2)$.

In order to obtain these approximations, we run Algorithm 5.1 for each complex embedding μ of L. The starting precision is chosen in

$$\widetilde{O}(\log \Delta_L + d\ell H + \ell B_{\Theta,\mu} + d^2),$$

for a suitable recipe of algebraic extensions Θ. Therefore, the cost to compute the required complex approximations in the embedding μ is

$$\widetilde{O}(\ell^2 B_{\Theta,\mu} + \ell \log \Delta_L + d\ell^2 H + \ell d^2)$$

binary operations. The sum of the bounds $B_{\Theta,\mu}$ is in $O(dH)$, hence the total cost over all embeddings is $\widetilde{O}(\ell^2 d^2 H + \ell d \log \Delta_L + \ell d^3)$ binary operations. □

These evaluation algorithms for modular equations can be modified to output derivatives of modular equations as well, as in Theorems 1.1 and 1.2, for the same asymptotic cost. Indeed, the algorithm of [27] actually computes derivatives of theta constants as well; moreover, Q_ℓ^2 and Q_β^2 can be used as denominators for derivatives of modular equations. Algorithm 5.1 can be formally differentiated to compute analytic approximations of derivatives of modular equations from derivatives of theta constants. One can check that the precision losses taken in the resulting algorithm remain within the asymptotic bounds given in Proposition 5.2.

References

[1] C. Birkenhake and H. Lange. *Complex abelian varieties*. Springer, second edition, 2004.

[2] C. Birkenhake and H. Wilhelm. Humbert surfaces and the Kummer plane. *Trans. Amer. Math. Soc.*, 335(5):1819–1841, 2003.

[3] C. W. Borchardt. Theorie des arithmetisch-geometrisches Mittels aus vier Elementen. In *Gesammelte Werke*, pages 373–431. Reimer, 1888.

[4] J.-B. Bost and J.-F. Mestre. Moyenne arithmético-géométrique et périodes de courbes de genre 1 et 2. *Gaz. Math.*, 38:36–64, 1988.

[5] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, and E. Schost. *Algorithmes efficaces en calcul formel*. CreateSpace, 2017.

[6] A. Bostan, F. Morain, B. Salvy, and É. Schost. Fast algorithms for computing isogenies between elliptic curves. *Math. Comp.*, 77(263):1755–1778, 2008.

[7] R. Bröker and K. Lauter. Modular polynomials for genus 2. *LMS J. Comp. Math.*, 12:326–339, 2009.

[8] R. Bröker, K. Lauter, and A. V. Sutherland. Modular polynomials via isogeny volcanoes. *Math. Comp.*, 81:1201–1231, 2012.

[9] E. Costa, N. Mascot, J. Sijsling, and J. Voight. Rigorous computation of the endomorphism ring of a Jacobian. *Math. Comp.*, 88(317):1303–1339, 2019.

[10] J.-M. Couveignes and T. Ezome. Computing functions on Jacobians and their quotients. *Lond. Math. Soc. J. Comput. Math.*, 18(1):555–577, 2015.

[11] D. A. Cox. *Primes of the form $x^2 + ny^2$*. John Wiley & Sons, second edition, 2013.

[12] L. Dembélé and J. Voight. Explicit methods for Hilbert modular forms. In *Elliptic Curves, Hilbert Modular Forms and Galois Deformations*, pages 135–198. Birkhäuser, 2013.

[13] A. Dudeanu, D. Jetchev, D. Robert, and M. Vuille. Cyclic isogenies for abelian varieties with real multiplication. 2017.

[14] R. Dupont. *Moyenne arithmético-géométrique, suites de Borchardt et applications*. PhD thesis, École polytechnique, 2006.

[15] N. D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational perspectives on number theory (Chicago, 1995)*, volume 7, pages 21–76. Amer. Math. Soc., 1998.

[16] A. Enge. Computing modular polynomials in quasi-linear time. *Math. Comp.*, 78(267):1809–1824, 2009.

[17] G. Faltings and C.-L. Chai. *Degeneration of abelian varieties*. Springer, 1990.

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

[19] W. Hart. FLINT: Fast Library for Number Theory. https://flintlib.org.

[20] M. Hindry and J. H. Silverman. *Diophantine Geometry*. Springer, 2000.

[21] J.-I. Igusa. On Siegel modular forms of genus two. *Amer. J. Math.*, 84:175–200, 1962.

[22] J.-I. Igusa. On the graded ring of theta-constants. *Amer. J. Math.*, 86(1):219–246, 1964.

[23] J.-I. Igusa. On the ring of modular forms of degree two over $\mathbb{Z}$. *Amer. J. Math.*, 101(1):149–183, 1979.

[24] F. Jarvis. Higher genus arithmetic-geometric means. *Ramanujan J.*, 17(1):1–17, 2008.

[25] F. Johansson. Arb: efficient arbitrary-precision midpoint-radius interval arithmetic. *IEEE Trans. Comput.*, 66(8):1281–1292, 2017.

[26] J. Kieffer. HDME: a C library for the evaluation of modular equations in dimension 2. https://github.com/j-kieffer/hdme.

[27] J. Kieffer. Certified Newton schemes for the evaluation of low-genus theta functions. 2022.

[28] J. Kieffer. Counting points on abelian surfaces over finite fields with Elkies's method. 2022.

[29] J. Kieffer. Degree and height estimates for modular equations on PEL Shimura varieties. *J. London Math. Soc.*, 2022.

[30] J. Kieffer. Sign choices in the AGM for genus two theta constants. *Pub. Math. Besançon*, to appear.

[31] J. Kieffer, A. Page, and D. Robert. Computing isogenies from modular equations in genus two. 2019.

[32] H. Klingen. *Introductory lectures on Siegel modular forms.* Cambridge University Press, 1990.

[33] H. Labrande and E. Thomé. Computing theta functions in quasi-linear time in genus 2 and above. In *Algorithmic Number Theory Symposium XII*, Kaiserslautern, 2016. *LMS J. Comp. Math*, 19:163–177.

[34] K. Lauter and T. Yang. Computing genus 2 curves from invariants on the Hilbert moduli space. *J. Number Theory*, 131(5):936–958, 2011.

[35] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

[36] D. Lubicz and D. Robert. Computing separable isogenies in quasi-optimal time. *LMS J. Comp. Math.*, 18(1):198–216, 2015.

[37] C. Martindale. Hilbert modular polynomials. *J. Number Theory*, 213:464–498, 2020.

[38] J.-F. Mestre. Construction de courbes de genre 2 à partir de leurs modules. In *Effective methods in algebraic geometry (Castiglioncello, 1990)*, page 313–334. Birkhäuser, 1991.

[39] E. Milio. Database of modular polynomials of Hilbert and Siegel. https://members.loria.fr/EMilio/modular-polynomials.

[40] E. Milio. A quasi-linear time algorithm for computing modular polynomials in dimension 2. *LMS J. Comput. Math.*, 18:603–632, 2015.

[41] E. Milio and D. Robert. Modular polynomials on Hilbert surfaces. *J. Number Theory*, 216:403–459, 2020.

[42] J. S. Milne. Introduction to Shimura varieties. In *Harmonic analysis, the trace formula, and Shimura varieties*, pages 265–378. Amer. Math. Soc., 2005.

[43] D. Mumford. *Tata lectures on theta. I.* Birkhäuser, 1983.

[44] D. Mumford. *Tata lectures on theta. II.* Birkhäuser, 1984.

[45] S. Nagaoka. On the ring of Hilbert modular forms over $\mathbb{Z}$. *J. Math. Soc. Japan*, 35(4):589–608, 1983.

[46] P. Nguyen and B. Vallée, editors. *The LLL algorithm. Survey and applications.* Springer, 2009.

[47] A. Novocin, D. Stehlé, and G. Villard. An LLL-reduction algorithm with quasi-linear time complexity. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, pages 403–412, San Jose, 2011. ACM.

[48] H. L. Resnikoff. On the graded ring of Hilbert modular forms associated with $\mathbb{Q}(\sqrt{5})$. *Math. Ann.*, 208:161–170, 1974.

[49] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théorie Nr. Bordx.*, 7(1):219–254, 1995.

[50] M. Streng. *Complex multiplication of abelian surfaces.* PhD thesis, Universiteit Leiden, 2010.

[51] M. Streng. Computing Igusa class polynomials. *Math. Comp.*, 83:275–309, 2014.

[52] A. V. Sutherland. On the evaluation of modular polynomials. In *Proceedings of the 10th Algorithmic Number Theory Symposium*, pages 531–555, San Diego, 2013. Math. Sci. Publ.

[53] W. Tucker. *Validated numerics: a short introduction to rigorous computations.* Princeton University Press, 2011.

[54] G. van der Geer. *Hilbert modular surfaces.* Springer, 1988.

[55] G. van der Geer. Siegel modular forms and their applications. In *The 1-2-3 of modular forms*, pages 181–245. Springer, 2008.

[56] J. Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris*, A273:238–241, 1971.

[57] B. Williams. The rings of Hilbert modular forms for $\mathbb{Q}(\sqrt{29})$ and $\mathbb{Q}(\sqrt{37})$. *J. Algebra*, 559(1):679–711, 2020.

[58] C. K. Yap. Fast unimodular reduction: planar integer lattices. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 436–446, Pittsburgh, 1992. IEEE.