CLASS INVARIANTS BY THE CRT METHOD

ANDREAS ENGE AND ANDREW V. SUTHERLAND

Abstract. We adapt the CRT approach for computing Hilbert class polynomials to handle a wide range of class invariants. For suitable discriminants D, this improves its performance by a large constant factor, more than 200 in the most favourable circumstances. This has enabled record-breaking constructions of elliptic curves via the CM method, including examples with $|D| > 10^{15}$.

1. Introduction

Every ordinary elliptic curve E over a finite field $\mathbb{F}_q$ has *complex multiplication* by an imaginary quadratic order $\mathcal{O}$, by which we mean that the endomorphism ring $\mathrm{End}(E)$ is isomorphic to $\mathcal{O}$. The Deuring lifting theorem implies that E is the reduction of an elliptic curve $\hat{E}/\mathbb{C}$ that also has complex multiplication by $\mathcal{O}$. Let K denote the fraction field of $\mathcal{O}$. The j-invariant of $\hat{E}$ is an algebraic integer whose minimal polynomial over K is the *Hilbert class polynomial* H_D, where D is the discriminant of $\mathcal{O}$. Notably, the polynomial H_D actually lies in $\mathbb{Z}[X]$, and its splitting field is the *ring class field* $K_{\mathcal{O}}$ for the order $\mathcal{O}$.

Conversely, an elliptic curve $E/\mathbb{F}_q$ with complex multiplication by $\mathcal{O}$ exists whenever q satisfies the norm equation $4q = t^2 - v^2 D$, with $t, v \in \mathbb{Z}$ and $t \not\equiv 0$ modulo the characteristic of $\mathbb{F}_q$. In this case H_D splits completely over $\mathbb{F}_q$, and its roots are precisely the j-invariants of the elliptic curves $E/\mathbb{F}_q$ that have complex multiplication by $\mathcal{O}$. Such a curve has $q + 1 \pm t$ points, where t is determined, up to a sign, by the norm equation. With a judicious selection of D and q one may obtain a curve with prescribed order. This is known as the *CM method*.

The main challenge for the CM method is to obtain the polynomial H_D, which has degree equal to the class number $h(D)$, and total size $O(|D|^{1+\epsilon})$. There are three approaches to computing H_D, all of which, under reasonable assumptions, can achieve a running time of $O(|D|^{1+\epsilon})$. These include the complex analytic method [12], a p-adic algorithm [9, 7], and an approach based on the Chinese Remainder Theorem (CRT) [2]. The first is the most widely used, and it is quite efficient; the range of discriminants to which it may be applied is limited not by its running time, but by the space required. The polynomial H_D is already likely to exceed available memory when $|D| > 10^9$, hence one seeks to apply the CM method to alternative class polynomials that have smaller coefficients than H_D. This makes computations with $|D| > 10^{10}$ feasible.

Recently, a modified version of the CRT approach was proposed that greatly reduces the space required for the CM method [30]. Under the Generalised Riemann Hypothesis (GRH), this algorithm is able to compute $H_D \bmod P$ using

$O(|D|^{1/2+\epsilon}\log P)$ space and $O(|D|^{1+\epsilon})$ time. (Here and in the following, all complexity estimates refer to bit operations.) The reduced space complexity allows it to handle much larger discriminants, including examples with $|D| > 10^{13}$.

An apparent limitation of the CRT approach is that it depends on some specific features of the j-function. As noted in [2], this potentially precludes it from computing class polynomials other than H_D. The purpose of the present article is to show how these obstructions may be overcome, allowing us to apply the CRT method to many functions other than j, including two infinite families.

Subject to suitable constraints on D, we may then compute a class polynomial with smaller coefficients than H_D (by a factor of up to 72), and, in certain cases, with smaller degree (by a factor of 2). Remarkably, the actual running time with the CRT method is typically *better* than the size difference would suggest. Fewer CRT moduli are needed, and we may choose a subset for which the computation is substantially faster than on average.

We start §2 with a brief overview of the CRT method, and then describe a new technique to improve its performance, which also turns out to be crucial for certain class invariants. After discussing families of invariants in §3, we consider CRT-based approaches applicable to the different families and give a general algorithm in §4. Computational results and performance data appear in §5.

2. Hilbert class polynomials via the CRT

2.1. **The algorithm of Belding, Bröker, Enge, Lauter and Sutherland.** The basic idea of the CRT-based algorithm for Hilbert class polynomials is to compute H_D modulo many small primes p, and then lift its coefficients by Chinese remaindering to integers, or to their reductions modulo a large (typically prime) integer P, via the explicit CRT [4, Thm. 3.1]. The latter approach suffices for most applications, and while it does not substantially reduce the running time (the same number of small primes is required), it can be accomplished using only $O(|D|^{1/2+\epsilon}\log P)$ space with the method of [30, §6].

For future reference, we summarise the algorithm to compute $H_D \bmod p$ for a prime p that splits completely in the ring class field $K_{\mathcal{O}}$. Let $h = h(D)$.

Algorithm 1 (Computing $\boldsymbol{H_D \bmod p}$).

(1) *Find the j-invariant j_1 of an elliptic curve $E/\mathbb{F}_p$ with* $\mathrm{End}(E) \cong \mathcal{O}$.
(2) *Enumerate the other roots $j_2, \ldots, j_h$ of H_D* $\bmod p$.
(3) *Compute* $H_D(X) \bmod p = (X - j_1)\cdots(X - j_h)$.

The first step is achieved by varying j_1 (systematically or randomly) over the elements of $\mathbb{F}_p$ until it corresponds to a suitable curve; details and many practical improvements are given in [2, 30]. The third step is a standard building block of computer algebra. Our interest lies in Step 2.

2.2. **Enumerating the roots of** $H_D \bmod p$**.** The key idea in [2] leading to a quasi-linear complexity is to apply the Galois action of $\mathrm{Cl}(\mathcal{O}) \simeq \mathrm{Gal}(K_{\mathcal{O}}/K)$. The group $\mathrm{Cl}(\mathcal{O})$ acts on the roots of H_D, and when p splits completely in $K_{\mathcal{O}}$ there is a corresponding action on the set $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p) = \{j_1, \ldots, j_h\}$ containing the roots of $H_D \bmod p$. For an ideal class $[\mathfrak{a}]$ in $\mathrm{Cl}(\mathcal{O})$ and a j-invariant $j_i \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, let us write $[\mathfrak{a}]j_i$ for the image of j_i under the Galois action of $[\mathfrak{a}]$. We then have $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p) = \{[\mathfrak{a}]j_1 : [\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})\}$.

As in [30, §5], we use a polycyclic presentation defined by a sequence of ideals $\mathfrak{l}_1, \dots, \mathfrak{l}_m$ with prime norms $\ell_1, \dots, \ell_m$ whose classes generate $\mathrm{Cl}(\mathcal{O})$. The *relative order* r_k is the least positive integer for which $[\mathfrak{l}_k^{r_k}] \in \langle [\mathfrak{l}_1], \dots, [\mathfrak{l}_{k-1}] \rangle$. We may then uniquely write $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1}] \cdots [\mathfrak{l}_m^{e_m}]$, with $0 \le e_k < r_k$. To maximise performance, we use a presentation in which $\ell_1 < \cdots < \ell_m$, with each ℓ_k as small as possible subject to $r_k > 1$. Note that the relative order r_k divides the order n_k of $[\mathfrak{l}_k]$ in $\mathrm{Cl}(\mathcal{O})$, but for $k > 1$ we can (and often do) have $r_k < n_k$.

For each $j_i \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ and each $\mathcal{O}$-ideal $\mathfrak{l}$ of prime norm ℓ, the j-invariant $[\mathfrak{l}]j_i$ corresponds to an ℓ-isogenous curve, which we may obtain as a root of $\Phi_\ell(j_i, X)$, where $\Phi_\ell \in \mathbb{Z}[J, J_\ell]$ is the *classical modular polynomial* [31, §69]. The polynomial Φ_ℓ has the pair of functions $\big(j(z), j(\ell z)\big)$ as roots, and parameterises isogenies of degree ℓ.

Fixing an isomorphism $\mathrm{End}(E) \cong \mathcal{O}$, we let $\pi \in \mathcal{O}$ denote the Frobenius endomorphism. When the order $\mathbb{Z}[\pi]$ is maximal at ℓ, the univariate polynomial $\Phi_\ell(j_i, X) \in \mathbb{F}_p[X]$ has exactly two roots $[\mathfrak{l}]j_i$ and $[\bar{\mathfrak{l}}]j_i$ when ℓ splits in $\mathcal{O}$, and a single root $[\mathfrak{l}]j_i$ if ℓ is ramified [25, Prop. 23]. To simplify matters, we assume here that $\mathbb{Z}[\pi]$ is maximal at each ℓ_k, but this is not necessary, see [30, §4].

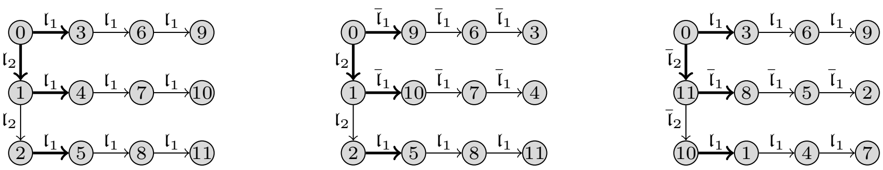
We may enumerate $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p) = \{[\mathfrak{a}]j_1 : [\mathfrak{a}] \in \langle [\mathfrak{l}_1], \dots, [\mathfrak{l}_m] \rangle\}$ via [30, Alg. 1.3]:

Algorithm 2 (Enumerating $\mathbf{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ — Step 2 of Algorithm 1)**.**

(1) *Let j_2 be an arbitrary root of $\Phi_{\ell_m}(j_1, X)$ in $\mathbb{F}_p$.*
(2) *For i from 3 to r_m, let j_i be the root of $\Phi_{\ell_m}(j_{i-1}, X)/(X - j_{i-2})$ in $\mathbb{F}_p$.*
(3) *If $m > 1$, then for i from 1 to r_m:*
Recursively enumerate the set $\{[\mathfrak{a}]j_i : [\mathfrak{a}] \in \langle [\mathfrak{l}_1], \dots, [\mathfrak{l}_{m-1}] \rangle\}$.

In general there are two distinct choices for j_2, but either will do. Once j_2 is chosen, $j_3, \dots, j_{r_m}$ are determined. The sequence $(j_1, \dots, j_{r_m})$ corresponds to a path of ℓ_m-isogenies; we call this path an ℓ_m-*thread*.

The choice of j_2 in Step 1 may change the order in which $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ is enumerated. Three of the sixteen possibilities when $m = 2$, $r_1 = 4$, and $r_2 = 3$ are shown below; we assume $[\mathfrak{l}_2^3] = [\mathfrak{l}_1]$, and label each vertex $[\mathfrak{l}_2^e]j_1$ by the exponent e.



Bold edges indicate where a choice was made. Regardless of these choices, Algorithm 2 correctly enumerates $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ in every case [30, Prop. 5].

2.3. **Finding roots with greatest common divisors (gcds).** The potentially haphazard manner in which Algorithm 2 enumerates $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ is not a problem when computing H_D, but it can complicate matters when we wish to compute other class polynomials. We could distinguish the actions of $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ using an Elkies kernel polynomial [10], as suggested in [7, §5], however this slows down the algorithm significantly. An alternative approach using polynomial gcds turns out to be much more efficient, and actually speeds up Algorithm 2, making it already a useful improvement when computing H_D.

We need not distinguish the actions of $\mathfrak{l}$ and $\bar{\mathfrak{l}}$ at this stage, but we wish to ensure that our enumeration of $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$ makes a consistent choice of direction each time it starts an ℓ-thread. The first ℓ-thread may be oriented arbitrarily, but for each subsequent ℓ-thread $(j_1', j_2', \ldots, j_r')$, we apply Lemma 1 below. This allows us to "square the corner" by choosing j_2' as the unique common root of $\Phi_\ell(X, j_1')$ and $\Phi_{\ell'}(X, j_2)$, where $(j_1, \ldots, j_r)$ is a previously computed ℓ-thread and j_1 is ℓ'-isogenous to j_1'. The edge (j_1, j_1') lies in an ℓ'-thread that has already been computed, for some $\ell' > \ell$.

$$\begin{tikzcd} j_1 \arrow[r, "\mathfrak{l}"] \arrow[d, "\mathfrak{l}'"'] & j_2 \arrow[r, "\mathfrak{l}"] \arrow[d, dashed, "\mathfrak{l}'"'] & j_3 \arrow[r, "\mathfrak{l}"] & \cdots \arrow[r, "\mathfrak{l}"] & j_r \\ j_1' \arrow[r, dashed, "\mathfrak{l}"] & j_2' \end{tikzcd} \qquad \begin{tikzcd} j_1 \arrow[r, "\mathfrak{l}"] \arrow[d, "\mathfrak{l}'"'] & j_2 \arrow[r, "\mathfrak{l}"] \arrow[d, dashed, "\mathfrak{l}'"'] & j_3 \arrow[r, "\mathfrak{l}"] \arrow[d, dashed, "\mathfrak{l}'"'] & \cdots \arrow[r, "\mathfrak{l}"] & j_r \arrow[d, dashed, "\mathfrak{l}'"'] \\ j_1' \arrow[r, dashed, "\mathfrak{l}"] & j_2' \arrow[r, dashed, "\mathfrak{l}"] & j_3' \arrow[r, dashed, "\mathfrak{l}"] & \cdots \arrow[r, dashed, "\mathfrak{l}"] & j_r' \end{tikzcd}$$

Having computed j_2', we could compute $j_3', \ldots, j_r'$ as before, but it is usually better to continue using gcds, as depicted above. Asymptotically, both root-finding and gcd computations are dominated by the $O(\ell^2 \mathsf{M}(\log p))$ time it takes to instantiate $\Phi_\ell(X, j_i)$ mod p, but in practice ℓ is small, and we effectively gain a factor of $O(\log p)$ by using gcds when $\ell \approx \ell'$. This can substantially reduce the running time of Algorithm 2, as may be seen in Table 1 of §5.

With the gcd approach described above, the total number of root-finding operations can be reduced from $\prod_{k=1}^m r_k$ to $\sum_{k=1}^m r_k$. When m is large, this is a big improvement, but it is no help when $m = 1$, as necessarily occurs when $h(D)$ is prime. However, even in this case we can apply gcds by looking for an auxiliary ideal $\mathfrak{l}_1'$, with prime norm ℓ_1', for which $[\mathfrak{l}_1'] = [\mathfrak{l}_1^e]$. When r_1 is large, such an $\mathfrak{l}_1'$ is easy to find, and we may choose the best combination of ℓ_1' and e available. This idea generalises to ℓ_k-threads, where we seek $[\mathfrak{l}_k'] \in \langle [\mathfrak{l}_1] \ldots, [\mathfrak{l}_k] \rangle \backslash \langle [\mathfrak{l}_1] \ldots, [\mathfrak{l}_{k-1}] \rangle$.

Lemma 1. *Let $j_1, j_2 \in \mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, and let $\ell_1, \ell_2 \neq p$ be distinct primes with $4\ell_1^2\ell_2^2 < |D|$. Then $\gcd\bigl(\Phi_{\ell_1}(j_1, X), \Phi_{\ell_2}(j_2, X)\bigr)$ has degree at most 1.*

Proof. It follows from [25, Prop. 23] that $\Phi_{\ell_1}(X, j_1)$ and $\Phi_{\ell_2}(X, j_2)$ have at most two common roots in the algebraic closure $\overline{\mathbb{F}}_p$, which in fact lie in $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$. If there are exactly two, then both $\ell_1 = \mathfrak{l}_1\bar{\mathfrak{l}}_1$ and $\ell_2 = \mathfrak{l}_2\bar{\mathfrak{l}}_2$ split in $\mathcal{O}$, and one of $\mathfrak{l}_1^2\mathfrak{l}_2^2$ or $\mathfrak{l}_1^2\bar{\mathfrak{l}}_2^2$ is principal with a non-rational generator. We thus have a norm equation $4\ell_1^2\ell_2^2 = a^2 - b^2 D$ with $a, b \in \mathbb{Z}$ and $b \neq 0$, and the lemma follows. $\square$

3. Class invariants

Due to the large size of H_D, much effort has been spent seeking smaller generators of $K_{\mathcal{O}}$. For a modular function f and $\mathcal{O} = \mathbb{Z}[\tau]$, with τ in the upper half plane, we call $f(\tau)$ a *class invariant* if $f(\tau) \in K_{\mathcal{O}}$. The *class polynomial* for f is

$$H_D[f](X) = \prod_{[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})} \left(X - [\mathfrak{a}]f(\tau)\right).$$

The contemporary tool for determining class invariants is Shimura's reciprocity law; see [28, Th. 4] for a fairly general result. Class invariants arising from many different modular functions have been described in the literature; we briefly summarise some of the most useful ones.

Let η be Dedekind's function, and let $\zeta_n = \exp(2\pi i/n)$. Weber considered

$$\mathfrak{f} = \zeta_{48}^{-1} \frac{\eta\left(\frac{z+1}{2}\right)}{\eta(z)}, \qquad \mathfrak{f}_1(z) = \frac{\eta\left(\frac{z}{2}\right)}{\eta(z)}, \qquad \mathfrak{f}_2(z) = \sqrt{2}\,\frac{\eta(2z)}{\eta(z)},$$

powers of which yield class invariants when $\left(\frac{D}{2}\right) \neq -1$, and also $\gamma_2 = \sqrt[3]{j}$, which is a class invariant whenever $3 \nmid D$. The Weber functions can be generalised [15, 16, 21, 20, 23], and we have the simple and double η-quotients

$$\mathfrak{w}_N(z) = \frac{\eta\left(\frac{z}{N}\right)}{\eta(z)}; \qquad \mathfrak{w}_{p_1,p_2} = \frac{\eta\left(\frac{z}{p_1}\right)\eta\left(\frac{z}{p_2}\right)}{\eta\left(\frac{z}{p_1p_2}\right)\eta(z)} \text{ with } N = p_1p_2,$$

where p_1 and p_2 are primes. Subject to constraints on D, including that no prime dividing N is inert in $\mathcal{O}$, suitable powers of these functions yield class invariants, see [15, 16]. For $s = 24/\gcd\bigl(24, (p_1-1)(p_2-1)\bigr)$, the canonical power $\mathfrak{w}_{p_1,p_2}^s$ is invariant under the Fricke involution $W|_N : z \mapsto \frac{-N}{z}$ for $\Gamma^0(N)$, equivalently, the Atkin-Lehner involution of level N, by [17, Thm. 2].

The theory of [28] applies to any functions for $\Gamma^0(N)$, in particular to those of prime level N invariant under the Fricke involution, which yield class invariants when $\left(\frac{D}{N}\right) \neq -1$. Atkin developed a method to compute such functions A_N, which are conjectured to have a pole of minimal order at the unique cusp [10, 26]. These are used in the SEA algorithm, and can be found in Magma or Pari/GP.

The functions above all yield algebraic integers, so $H_D[f] \in \mathcal{O}_K[X]$. Except for $\mathfrak{w}_N^e$ or when $\gcd(N, D) \neq 1$, in which cases additional restrictions may apply, one actually has $H_D[f] \in \mathbb{Z}[X]$, cf. [16, Cor. 3.1]. The (logarithmic) *height* of $H_D[f] = \sum a_i X^i$ is $\log \max |a_i|$, which determines the precision needed to compute the a_i. We let $c_D(f)$ denote the ratio of the heights of $H_D[j]$ and $H_D[f]$.

With $c(f) = \lim_{|D|\to\infty} c_D(f)$, we have: $c(\gamma_2) = 3$; $c(\mathfrak{f}) = 72$ (when $\left(\frac{D}{2}\right) = 1$);

$$c(\mathfrak{w}_N^e) = \frac{24(N+1)}{e(N-1)}; \qquad c(\mathfrak{w}_{p_1,p_2}^s) = \frac{12\psi(p_1p_2)}{s(p_1-1)(p_2-1)}; \qquad c(A_N) = \frac{N+1}{2|v_N|},$$

where e divides the exponent s defined above, v_N is the order of the pole of A_N at the cusp, and $\psi(p_1p_2)$ is $(p_1+1)(p_2+1)$ when $p_1 \neq p_2$, and $p_1(p_1+1)$ when $p_1 = p_2$. Morain observed in [27] that $c(A_{71}) = 36$, which is so far the best value known when $\left(\frac{D}{2}\right) = -1$. We conjecture that in fact for all primes $N > 11$ with $N \equiv 11 \bmod 60$ we have $c(A_N) = 30\frac{N+1}{N-11}$, and that for $N \equiv -1 \bmod 60$ we have $c(A_N) = 30$. This implies that given an arbitrary discriminant D, we can always choose N so that A_N yields class invariants with $c_D(A_N) \geq 30 + o(1)$.

When the prime divisors of N are all ramified in K, both $\mathfrak{w}_{p_1,p_2}$ and A_N yield class polynomials that are squares in $\mathbb{Z}[X]$, see [11, §1.6] and [18]. Taking the square root of such a class polynomial reduces both its degree and its height by a factor of 2. For a composite fundamental discriminant D (the most common case), this applies to $H_D[A_N]$ for any prime $N \mid D$. In the best case, D is divisible by 71, and we obtain a class polynomial that is 144 times smaller than H_D.

3.1. **Modular polynomials.** Each function $f(z)$ considered above is related to $j(z)$ by a modular polynomial $\Psi_f \in \mathbb{Z}[F, J]$ satisfying $\Psi_f(f(z), j(z)) = 0$. For primes ℓ not dividing the level N, we let $\Phi_{\ell,f}$ denote the minimal polynomial satisfying $\Phi_{\ell,f}(f(z), f(\ell z)) = 0$; it is a factor of $\operatorname{Res}_{J_\ell}\bigl(\operatorname{Res}_J(\Phi_\ell(J, J_\ell), \Psi_f(F, J)), \Psi_f(F_\ell, J_\ell)\bigr)$, and as such, an element of $\mathbb{Z}[F, F_\ell]$. Thus $\Phi_{\ell,f}$ generalises the classical modular polynomial $\Phi_\ell = \Phi_{\ell,j}$.

The polynomial $\Phi_{\ell,f}$ has degree $d(\ell+1)$ in F and F_ℓ, where d divides $\deg_J \Psi_f$, see [6, §6.8], and $2d$ divides $\deg_J \Psi_f$ when f is invariant under the Fricke involution. In general, d is maximal, and $d = 1$ is achievable only in the relatively few cases where $X_0(N)$, respectively $X_0^+(N)$, is of genus 0 and, moreover, f is a hauptmodul, that is, it generates the function field of the curve. Happily, this includes many cases of practical interest.

The polynomial Ψ_f characterises the analytic function f in an algebraic way; when $d = 1$, the polynomials Φ_ℓ and $\Phi_{\ell,f}$ algebraically characterise ℓ-isogenies between elliptic curves given by their j-invariants, or by class invariants derived from f, respectively. These are key ingredients for the CRT method.

4. CRT algorithms for class invariants

To adapt Algorithm 1 to class invariants arising from a modular function $f(z)$ other than $j(z)$, we only need to consider Algorithm 2. Our objective is to enumerate the roots of $H_D[f] \bmod p$ for suitable primes p, which we are free to choose. This may be done in one of two ways. The most direct approach computes an "f-invariant" f_1, corresponding to j_1, then enumerates $f_2, \ldots, f_h$ using the modular polynomials $\Phi_{\ell,f}$. Alternatively, we may enumerate $j_1, \ldots, j_h$ as before, and from these derive $f_1, \ldots, f_h$. The latter approach is not as efficient, but it applies to a wider range of functions, including two infinite families.

Several problems arise. First, an elliptic curve $E/\mathbb{F}_p$ with CM by $\mathcal{O}$ unambiguously defines a j-invariant $j_1 = j(E)$, but not the corresponding f_1. The f_1 we seek is a root of $\psi_f(X) = \Psi_f(X, j_1) \bmod p$, but ψ_f may have other roots, which may or may not be class invariants. The same problem occurs for the p-adic lifting algorithm and can be solved generically [6, §6]; we describe some more efficient solutions, which are in part specific to certain types of functions.

When ψ_f has multiple roots that are class invariants, these may be roots of distinct class polynomials. We are generally happy to compute any one of these, but it is imperative that we compute the reduction of "the same" class polynomial $H_D[f]$ modulo each prime p.

The lemma below helps to address these issues for at least two infinite families of functions: the double η-quotients $\mathfrak{w}_{p_1,p_2}$ and the Atkin functions A_N.

Lemma 2. *Let f be a modular function for $\Gamma^0(N)$, invariant under the Fricke involution $W|_N$, such that $f(z)$ and $f\left(\frac{-1}{z}\right)$ have rational q-expansions. Let the imaginary quadratic order $\mathcal{O}$ have conductor coprime to N and contain an ideal $\mathfrak{n} = \left(N, \frac{B_0+\sqrt{D}}{2}\right)$. Let $A_0 = \frac{B_0^2-D}{4N}$ and $\tau_0 = \frac{-B_0+\sqrt{D}}{2A_0}$, and assume that $\gcd(A_0, N) = 1$. Then $f(\tau_0)$ is a class invariant, and if $f(\tau)$ is any of its conjugates under the action of $\mathrm{Gal}(K_\mathcal{O}/K)$ we have*

$$\Psi_f\big(f(\tau), j(\tau)\big) = 0 \qquad \text{and} \qquad \Psi_f\big(f(\tau), [\mathfrak{n}]j(\tau)\big) = 0.$$

Proof. By definition, $\Psi_f\big(f(z), j(z)\big) = 0$. Applying the Fricke involution yields $0 = \Psi_f\left((W|_N f)(z), (W|_N j)(z)\right) = \Psi_f\left(f(z), j\left(\frac{-N}{z}\right)\right) = \Psi_f\left(f(z), j\left(\frac{z}{N}\right)\right)$. The value $f(\tau_0)$ is a class invariant by [28, Th. 4]. By the same result, we may assume that τ is the basis quotient of an ideal $\mathfrak{a} = \left(A, \frac{-B+\sqrt{D}}{2}\right)$ with $\gcd(A, N) = 1$ and $B \equiv B_0 \bmod 2N$. Then $\frac{\tau}{N}$ is the basis quotient of $\mathfrak{a}\overline{\mathfrak{n}} = \left(AN, \frac{-B+\sqrt{D}}{2}\right)$. It follows that $[\mathfrak{n}]j(\tau) = j\left(\frac{\tau}{N}\right)$, and replacing z above by τ completes the proof. □

If we arrange the roots of H_D into a graph of $\mathfrak{n}$-isogeny cycles corresponding to the action of $\mathfrak{n}$, the lemma yields a dual graph defined on the roots of $H_D[f]$, in which vertices $f(\tau)$ correspond to edges $\big(j(\tau), [\mathfrak{n}]j(\tau)\big)$.

In computational terms, $f(\tau)$ is a root of $\gcd\big(\Psi_f\big(X, j(\tau)\big), \Psi_f\big(X, [\mathfrak{n}]j(\tau)\big)\big)$. Generically, we expect this gcd to have no other roots modulo primes p that split completely in $K_{\mathcal{O}}$. For a finite number of such primes, there may be additional roots. We have observed this for p dividing the conductor of the order generated by $f(\tau)$ in the maximal order of $K_{\mathcal{O}}$. Such primes may either be excluded from our CRT computations, or addressed by one of the techniques described in §4.3.

4.1. **Direct enumeration.** When the polynomials $\Phi_{\ell,f}$ have degree $\ell+1$ we can apply Algorithm 2 with essentially no modification; the only new consideration is that ℓ must not divide the level N, but we can exclude such ℓ when choosing a polycyclic presentation for $\mathrm{Cl}(\mathcal{O})$. When the degree is greater than $\ell+1$ the situation is more complex, moreover the most efficient algorithms for computing modular polynomials do not apply [8, 13], making it difficult to obtain $\Phi_{\ell,f}$ unless ℓ is very small. Thus in practice we do not use $\Phi_{\ell,f}$ in this case; instead we apply the methods of §4.3 or §4.4. For the remainder of this subsection and the next we assume that we do have polynomials $\Phi_{\ell,f}$ of degree $\ell+1$ with which to enumerate $f_1,\ldots,f_h$, and consider how to determine a starting point f_1, given the j-invariant $j_1 = j(E)$ of an elliptic curve $E/\mathbb{F}_p$ with CM by $\mathcal{O}$.

When $\psi_f(X) = \Psi_f(X, j_1) \bmod p$ has only one root, our choice of f_1 is immediately determined. This is usually not the case, but we may be able to ensure it by restricting our choice of p. As an example, for $f = \gamma_2$ with $3 \nmid D$, if we require that $p \equiv 2 \bmod 3$, then f_1 is the unique cube root of j_1 in $\mathbb{F}_p$. If we additionally have $D \equiv 1 \bmod 8$ and $p \equiv 3 \bmod 4$, then the equation $\gamma_2 = (\mathfrak{f}^{24} - 16)/\mathfrak{f}^8$ uniquely determines the square of the Weber $\mathfrak{f}$ function, by [8, Lem. 7.3]. To treat $\mathfrak{f}$ itself we need an additional trick described in §4.2.

The next simplest case occurs when only one of the roots of ψ_f is a class invariant. This necessarily happens when f is invariant under the Fricke involution and all the primes dividing N are ramified in $\mathcal{O}$. In the context of Lemma 2, each root of $H_D[f]$ then corresponds to an isolated edge $\big(j(\tau), [\mathfrak{n}]j(\tau)\big)$ in the $\mathfrak{n}$-isogeny graph on the roots of H_D, and we compute f_1 as the unique root of $\gcd\big(\Psi_f(X, j_1), \Psi_f(X, [\mathfrak{n}]j_1)\big)$. In this situation $\mathfrak{n} = \bar{\mathfrak{n}}$, and each $f(\tau)$ occurs twice as a root of $H_D[f]$. By using a polycyclic presentation for $\mathrm{Cl}(\mathcal{O})/\langle[\mathfrak{n}]\rangle$ rather than $\mathrm{Cl}(\mathcal{O})$, we enumerate each double root of $H_D[f] \bmod p$ just once.

Even when ψ_f has multiple roots that are class invariants, it may happen that they are all roots of the *same* class polynomial. This applies to the Atkin functions $f = A_N$. When N is a split prime, there are two N-isogenous pairs $(j_1, [\mathfrak{n}]j_1)$ and $([\bar{\mathfrak{n}}]j_1, j_1)$ in $\mathrm{Ell}_{\mathcal{O}}(\mathbb{F}_p)$, and under Lemma 2 these correspond to roots f_1 and $[\bar{\mathfrak{n}}]f_1$ of ψ_f. Both are roots of $H_D[f]$, and we may choose either.

The situation is slightly more complicated for the double η-quotients $\mathfrak{w}_{p_1,p_2}$, with $N = p_1p_2$ composite. If $p_1 = \mathfrak{p}_1\bar{\mathfrak{p}}_1$ and $p_2 = \mathfrak{p}_2\bar{\mathfrak{p}}_2$ both split and $p_1 \neq p_2$, then there are four distinct N-isogenies corresponding to four roots of ψ_f. Two of these roots are related by the action of $[\mathfrak{n}] = [\mathfrak{p}_1\mathfrak{p}_2]$; they belong to the same class polynomial, which we choose as $H_D[f] \bmod p$. The other two are related by $[\mathfrak{p}_1\bar{\mathfrak{p}}_2]$ and are roots of a different class polynomial. We make an arbitrary choice for f_1, explicitly compute $[\mathfrak{n}]f_1$, and then check whether it occurs among the other three roots; if not, we correct the initial choice. The techniques of §4.3 may be used to efficiently determine the action of $[\mathfrak{n}]$.

Listed below are some of the modular functions f for which the roots of $H_D[f] \bmod p$ may be directly enumerated, with sufficient constraints on D and p. In each case p splits completely in $K_{\mathcal{O}}$ and $D < -4N^2$ has conductor u.

(1) γ_2, with $3 \nmid D$ and $p \equiv 2 \bmod 3$;

(2) $\mathfrak{f}^2$, with $D \equiv 1 \bmod 8$, $3 \nmid D$, and $p \equiv 11 \bmod 12$;

(3) $\mathfrak{w}_N^s$, for $N \in \{3, 5, 7, 13\}$ and $s = 24/\gcd(24, N-1)$, with $N \mid D$ and $N \nmid u$;

(4) $\mathfrak{w}_5^2$, with $3 \nmid D$, $5 \mid D$, and $5 \nmid u$;

(5) A_N, for $N \in \{3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 41, 47, 59, 71\}$, with $\left(\frac{D}{N}\right) \neq -1$ and $N \nmid u$.

(6) $\mathfrak{w}_{p_1,p_2}^s$, for $(p_1, p_2) \in \{(2,3), (2,5), (2,7), (2,13), (3,5), (3,7), (3,13), (5,7)\}$ and $s = 24/\gcd\big(24, (p_1-1)(p_2-1)\big)$, with $\left(\frac{D}{p_1}\right), \left(\frac{D}{p_2}\right) \neq -1$ and $p_1, p_2 \nmid u$.

(7) $\mathfrak{w}_{3,3}^6$ with $\left(\frac{D}{3}\right) = 1$ and $3 \nmid u$.

4.2. **The trace trick.** In §4.1 we were able to treat the square of the Weber $\mathfrak{f}$ function but not $\mathfrak{f}$ itself. To remedy this, we generalise a method suggested to us by Reinier Bröker.

We consider the situation where there are two modular functions f and f' that are roots of $\Psi_f(X, j(z))$, both of which yield class invariants for $\mathcal{O}$, and we wish to apply the direct enumeration approach. We assume that p is chosen so that $\psi_f(X) = \Psi_f(X, j_1) \bmod p$ has exactly two roots, and depending on which root we take as f_1, we may compute the reduction of either $H_D[f](X)$ or $H_D[f'](X)$ modulo p. In the case of Weber $\mathfrak{f}$, we have $f' = -f$, and $H_D[f']$ differs from $H_D[f]$ only in the sign of every other coefficient.

Consider a fixed coefficient a_i of $H_D[f](X) = \sum a_i X^i$; most of the time, the trace $t = -a_{h-1} = f_1 + \cdots + f_h$ will do (if $f' = -f$, we need to use a_i with $i \not\equiv h \bmod 2$). The two roots f_1 and f_1' lead to two possibilities t and t' modulo p. However, the elementary symmetric functions $T_1 = t + t'$ and $T_2 = tt'$ are unambiguous modulo p. Computing these modulo many primes p yields T_1 and T_2 as integers (via the CRT), from which t and t' are obtained as roots of the quadratic equation $X^2 - T_1 X + T_2$. If these are different, we arbitrarily pick one of them, which, going back, determines the set of conjugates $\{f_1, \ldots, f_h\}$ or $\{f_1', \ldots, f_h'\}$ to take modulo each of the primes $p \nmid t - t'$. In the unlikely event that they are the same (the suspicion $t = t'$ being confirmed after, say, looking at the second prime), we need to switch to a different coefficient a_i.

If f and f' differ by a simple transformation (such as $f' = -f$), the second set of conjugates and the value t' are obtained essentially for free. As a special case, when h is odd and the class invariants are units (as with Weber $\mathfrak{f}$), we can simply fix $t = a_0 = 1$, and need not compute $T_1 = 0$ and $T_2 = -1$.

The key point is that the number of primes p we use to determine t is much less than the number of primes we use to compute $H_D[f]$. Asymptotically, the logarithmic height of the trace is smaller than the height bound we use for $H_D[f]$ by a factor quasi-linear in $\log |D|$, under the GRH. In practical terms, determining t typically requires less than one tenth of the primes used to compute $H_D[f]$, and these computations can be combined.

The approach described above generalises immediately to more than two roots, but this case does not occur for the functions we examine. Unfortunately it can be used only in conjunction with the direct enumeration approach of §4.1; otherwise we would have to consistently distinguish not only between f_1 and f_1', but also between f_i and f_i' for $i = 2, \ldots, h$.

4.3. **Enumeration via the Fricke involution.** For functions f to which Lemma 2 applies, we can readily obtain the roots of $H_D[f] \bmod p$ without using the polynomials $\Phi_{\ell,f}$. We instead enumerate the roots of $H_D \bmod p$ (using the polynomials Φ_ℓ), and arrange them into a graph G of $\mathfrak{n}$-isogeny cycles, where $\mathfrak{n}$ is the ideal of norm N appearing in Lemma 2. We then obtain roots of $H_D[f] \bmod p$ by computing $\gcd\big(\Psi_f(X, j_i), \Psi_f(X, [\mathfrak{n}]j_i)\big)$ for each edge $(j_i, [\mathfrak{n}]j_i)$ in G.

The graph G is composed of h/n cycles of length n, where n is the order of $[\mathfrak{n}]$ in $\mathrm{Cl}(\mathcal{O})$. We assume that the $\mathcal{O}$-ideals of norm N are all non-principal and inequivalent (by requiring $|D| > 4N^2$ if needed). When every prime dividing N is ramified in $\mathcal{O}$ we have $n = 2$; as noted in §4.1, every root of $H_D[f]$ then occurs with multiplicity 2, and we may compute the square-root of $H_D[f]$ by taking each root just once. Otherwise we have $n > 2$.

Let $[\mathfrak{l}_1], \ldots, [\mathfrak{l}_m]$ be a polycyclic presentation for $\mathrm{Cl}(\mathcal{O})$ with relative orders $r_1, \ldots, r_m$, as in §2.2. For k from 1 to m let us fix $\mathfrak{l}_k = \big(\ell_k, \frac{-B_k+\sqrt{D}}{2}\big)$ with $B_k \geq 0$. To each vector $\vec{e} = (e_1, \ldots, e_m)$ with $0 \leq e_k < r_k$, we associate a unique root $j_{\vec{e}}$ enumerated by Algorithm 2, corresponding to the path taken from j_1 to $j_{\vec{e}}$, where e_k counts steps taken along an ℓ_k-thread. For $\vec{o} = (0, \ldots, 0)$ we have $j_{\vec{o}} = j_1$, and in general

$$j_{\vec{e}} = [\mathfrak{l}_1^{\sigma_1 e_1} \cdots \mathfrak{l}_m^{\sigma_m e_m}] j_{\vec{o}},$$

with $\sigma_k = \pm 1$. Using the method of §2.3 to consistently orient the ℓ_k-threads ensures that each σ_k depends only on the orientation of the first ℓ_k-thread.

To compute the graph G we must determine the signs σ_k. For those $[\mathfrak{l}_k]$ of order 2, we let $\sigma_k = 1$. We additionally fix $\sigma_k = 1$ for the least $k = k_0$ (if any) for which $[\mathfrak{l}_k]$ has order greater than 2, since we need not distinguish the actions of $\mathfrak{n}$ and $\bar{\mathfrak{n}}$. It suffices to show how to determine σ_k, given that we know $\sigma_1, \ldots, \sigma_{k-1}$. We may assume $[\mathfrak{l}_{k_0}]$ and $[\mathfrak{l}_k]$ both have order greater than 2, with $k_0 < k \leq m$.

Let $\mathfrak{l}$ be an auxiliary ideal of prime norm ℓ such that $[\mathfrak{l}] = [\mathfrak{a}\mathfrak{b}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_k^{e_k}]$, with $0 \leq e_i < r_i$, where $\mathfrak{b} = \mathfrak{l}_k^{e_k}$, and $[\mathfrak{a}]$ and $[\mathfrak{b}]$ have order greater than 2. Our assumptions guarantee that such an $\mathfrak{l}$ exists, by the Čebotarev density theorem, and under the GRH, ℓ is relatively small [1]. The fact that $[\mathfrak{a}]$ and $[\mathfrak{b}]$ have order greater than 2 ensures that $[\mathfrak{a}\bar{\mathfrak{b}}]$ is distinct from $[\mathfrak{l}]$ and its inverse. It follows that $\sigma_k = 1$ if and only if $\Phi_\ell(j_{\vec{o}}, j_{\vec{e}}) = 0$, where $\vec{e} = (e_1, \ldots, e_k, 0, \ldots, 0)$.

Having determined the σ_k, we compute the unique vector $\vec{v} = (v_1, \ldots, v_m)$ for which $[\mathfrak{n}] = [\mathfrak{l}_1^{\sigma_1 v_1} \cdots \mathfrak{l}_m^{\sigma_m v_m}]$. We then have $[\mathfrak{n}]j_{\vec{o}} = j_{\vec{v}}$, yielding the edge $(j_{\vec{o}}, j_{\vec{v}})$ of G. In general, we obtain the vector corresponding to $[\mathfrak{n}]j_{\vec{e}}$ by computing $\vec{e} + \vec{v}$ and using relations $[\mathfrak{l}_k^{r_k}] = [\mathfrak{l}_1^{x_1} \cdots \mathfrak{l}_{k-1}^{x_{k-1}}]$ to reduce the result, cf. [30, §5].

This method may be used with any function f satisfying Lemma 2, and in particular it applies to two infinite families of functions:

(8) A_N, for $N > 2$ prime, with $\left(\frac{D}{N}\right) \neq -1$ and $N \nmid u$.

(9) $\mathfrak{w}^s_{p_1,p_2}$, for p_1, p_2 primes not both 2, with $\left(\frac{D}{p_1}\right), \left(\frac{D}{p_2}\right) \neq -1$ and $p_1, p_2 \nmid u$.

As above, u denotes the conductor of $D < -4N^2$.

As noted earlier, for certain primes p we may have difficulty computing the edges of G when $\gcd\big(\Psi_f(X, j_i), \Psi_f(X, [\mathfrak{n}]j_i)\big)$ has more than one root in $\mathbb{F}_p$. While we need not use such primes, it is often easy to determine the correct root. Here we give two heuristic techniques for doing so.

The first applies when N is prime, as with the Atkin functions. In this case problems can arise when $H_D[f]$ has repeated roots modulo p. By Kummer's criterion, this can happen only when p divides the discriminant of $H_D[f]$, and even then, a repeated root x_1 is only actually a problem when it corresponds to two alternating edges in G, say (j_1, j_2) and (j_3, j_4), with the edge (j_2, j_3) between them. In this scenario we will get two roots x_1 and x_2 of $\gcd\bigl(\Psi_f(X, j_2), \Psi_f(X, j_3)\bigr)$. But if we already know that x_1 corresponds to (j_1, j_2), we can unambiguously choose x_2. In each of the N-isogeny cycles of G, it is enough to find a single edge that yields a unique root. If no such edge exists, then every edge must yield the *same* two roots x_1 and x_2, and we count each with multiplicity $n/2$.

The second technique applies when the roots of $H_D[f]$ are units, as with the double η-quotients [16, Thm. 3.3]. The product of the roots is then ± 1. Assuming that the number of edges in G for which multiple roots arise is small (it is usually zero, and rarely more than one or two), we simply test all the possible choices of roots and see which yield ± 1. If only one combination works, then the correct choices are determined. This is not guaranteed to happen, but in practice it almost always does.

4.4. **A general algorithm.** We now briefly consider the case of an arbitrary modular function f of level N, and sketch a general algorithm to compute $H_D[f]$ with the CRT method.

Let us assume that $f(\tau)$ is a class invariant, and let D be the discriminant and u the conductor of the order $\mathcal{O} = [1, \tau]$. The roots of $\Psi_f(X, j(\tau)) \in K_{\mathcal{O}}[X]$ lie in the ray class field of conductor uN over K, and some number n of these, including $f(\tau)$, actually lie in the ring class field $K_{\mathcal{O}}$. We may determine n using the method described in [6, §6.4], which computes the action of $(\mathcal{O}/N\mathcal{O})^*/\mathcal{O}^*$ on the roots of $\Psi_f(X, j(\tau))$. We note that the complexity of this task is essentially fixed as a function of $|D|$.

Having determined n, we use Algorithm 2 to enumerate the roots $j_1, \ldots, j_h$ of $H_D \bmod p$ as usual, but if for any j_i we find that $\Psi_f(X, j_i) \bmod p$ does not have exactly n roots $f_i^{(1)}, \ldots, f_i^{(n)}$, we exclude the prime p from our computations. The number of such p is finite and may be bounded in terms of the discriminants of the polynomials $\Psi_f(X, \alpha)$ as α ranges over the roots of $H_D[f]$. We then compute the polynomial $H(X) = \prod_{i=1}^{h} \prod_{r=1}^{n} \bigl(X - f_i^{(r)}\bigr)$ of degree nh in $\mathbb{F}_p[X]$. After doing this for sufficiently many primes p, we can lift the coefficients by Chinese remaindering to the integers. The resulting H is a product of n distinct class polynomials, all of which may be obtained by factoring H in $\mathbb{Z}[X]$. Under suitable heuristic assumptions (including the GRH), the total time to compute $H_D[f]$ is quasi-linear in $|D|$, including the time to factor H.

This approach is practically efficient only when n is small, but then it can be quite useful. A notable example is the modular function g for which

$$\Psi_g(X, J) = (X^{12} - 6X^6 - 27)^3 - JX^{18}.$$

This function was originally proposed by Atkin, and is closely related to certain class invariants of Ramanujan [3, Thm. 4.1]. The function g yields class invariants when $D \equiv 13 \bmod 24$. In terms of our generic algorithm, we have $n = 2$, and for $p \equiv 2 \bmod 3$ we get exactly two roots of $\Psi_g(X, j_i) \bmod p$, which differ only in sign. Thus $H(X) = H_D[g^2](X^2) = H_D[g](X)H_D[g](-X)$, and from this we easily obtain $H_D[g^2]$, and also $H_D[g]$ if desired.

5. Computational Results

This section provides performance data for the techniques developed above. We used AMD Phenom II 945 CPUs clocked at 3.0 GHz for our tests; the software was implemented using the `gmp` [22] and `zn_poly` [24] libraries, and compiled with `gcc` [19].

To compute the class polynomial $H_D[f]$, we require a bound on the size of its coefficients. Unfortunately, provably accurate bounds for functions f other than j are generally unavailable. As a heuristic, we take the bound B on the coefficients of H_D given by [30, Lem. 8], divide $\log_2 B$ by the asymptotic height factor $c(f)$, and add a "safety margin" of 256 bits. We note that with the CM method, the correctness of the final result can be efficiently and unconditionally confirmed [5], so we are generally happy to work with a heuristic bound.

5.1. **Class polynomial computations using the CRT method.** Our first set of tests measures the improvement relative to previous computations with the CRT method. We used discriminants related to the construction of a large set of pairing-friendly elliptic curves, see [30, §8] for details. We reconstructed many of these curves, first using the Hilbert class polynomial H_D, and then using an alternative class polynomial $H_D[f]$. In each case we used the explicit CRT to compute H_D or $H_D[f]$ modulo a large prime q (170 to 256 bits).

Table 1 gives results for four discriminants with $|D| \approx 10^{10}$, three of which appear in [30, Table 2]. Each column lists times for three class polynomial computations. First, we give the total time T_{tot} to compute $H_D \bmod q$, including the time T_{enum} spent enumerating $\mathrm{Ell}_D(\mathbb{F}_p)$, for all the small primes p, using Algorithm 2 as it appears in §2.2. We then list the times T'_{enum} and T'_{tot} obtained when Algorithm 2 is modified to use gcd computations whenever it is advantageous to do so, as explained in §2.3. The gcd approach typically speeds up Algorithm 2 by a factor of 2 or more.

For the third computation we selected a function f that yields class invariants for D, and computed $H_D[f] \bmod q$. This polynomial can be used in place of H_D in the CM method (one extracts a root x_0 of $H_D[f] \bmod q$, and then extracts a root of $\Psi_f(x_0, J) \bmod q$). For each function f we give a "size factor", which approximates the ratio of the total size of H_D to $H_D[f]$ (over $\mathbb{Z}$). In the first three examples this is just the height factor $c(f)$, but in Example 4 it is $4c(f)$ because the prime 59 is ramified and we actually work with the square root of $H_D[A_{59}]$, as noted in §4.1, reducing both the height and degree by a factor of 2.

We then list the speedup $T'_{\text{tot}}/T'_{\text{tot}}[f]$ attributable to computing $H_D[f]$ rather than H_D. Remarkably, in each case this speedup is about twice what one would expect from the height factor. This is explained by a particular feature of the CRT method: The cost of computing $H_D \bmod p$ for small primes p varies significantly, and, as explained in [30, §3], one can accelerate the CRT method with a careful choice of primes. When fewer small primes are needed, we choose those for which Step 1 of Algorithm 1 can be performed most quickly.

The last line in Table 1 lists the total speedup $T_{\text{tot}}/T'_{\text{tot}}[f]$ achieved.

5.2. **Comparison to the complex analytic method.** Our second set of tests compares the CRT approach to the complex analytic method. For each of the five discriminants listed in Table 2 we computed class polynomials $H_D[f]$ for the double η-quotient $\mathfrak{w}_{3,13}$ and the Weber $\mathfrak{f}$ function, using both the CRT approach described here, and the implementation [14] of the complex analytic method as described in

	Example 1	Example 2	Example 3	Example 4
$\|D\|$	13569850003	11039933587	12901800539	12042704347
$h(D)$	20203	11280	54706	9788
$\lceil \log_2 B \rceil$	2272564	1359134	5469776	1207412
$(\ell_1^{r_1}, \ldots, \ell_k^{r_k})$	(7^{20203})	$(17^{1128}, 19^{10})$	$(3^{27038}, 5^2)$	$(29^{2447}, 31^2, 43^2)$
T_{enum} (roots)	6440	10200	10800	21700
T_{tot}	19900	23700	52200	42400
T'_{enum} (gcds)	2510	2140	3440	4780
T'_{tot}	15900	15500	44700	25300
Function f	A_{71}	A_{47}	A_{71}	A_{59}
Size factor	36	24	36	120*
$T'_{\text{tot}}[f]$	213	305	629	191
Speedup $(T'_{\text{tot}}/T'_{\text{tot}}[f])$	75	51	71	132
Speedup $(T_{\text{tot}}/T'_{\text{tot}}[f])$	**93**	**78**	**83**	**222**

Table 1. Example class polynomial computations (times in CPU seconds)

		complex analytic		CRT		CRT mod q	
$\|D\|$	$h(D)$	$\mathfrak{w}_{3,13}$	$\mathfrak{f}$	$\mathfrak{w}_{3,13}$	$\mathfrak{f}$	$\mathfrak{w}_{3,13}$	$\mathfrak{f}$
6961631	5000	15	5.4	2.2	1.0	2.1	1.0
23512271	10000	106	33	10	4.1	9.8	4.0
98016239	20000	819	262	52	22	47	22
357116231	40000	6210	1900	248	101	213	94
2093236031	100000	91000	27900	2200	870	1800	770

Table 2. CRT vs. complex analytic (times in CPU seconds)

[12]. With the CRT we computed $H_D[f]$ both over $\mathbb{Z}$ and modulo a 256-bit prime q; for the complex analytic method these times are essentially the same.

We also tested a "worst case" scenario for the CRT approach: the discriminant $D = -85702502803$, for which the smallest non-inert prime is $\ell_1 = 109$. Choosing the function most suitable to each method, the complex analytic method computes $H_D[\mathfrak{w}_{109,127}]$ in 8310 seconds, while the CRT method computes $H_D[A_{131}]$ in 7150 seconds. The CRT approach benefits from the attractive height factor of the Atkin functions, $c(A_{131}) = 33$ versus $c(\mathfrak{w}_{109,127}) \approx 12.4$, and the use of gcds in Algorithm 2. Without these improvements, the time to compute H_D with the CRT method is 1460000 seconds. The techniques presented here yield more than a 200-fold speedup in this example.

5.3. **A record-breaking CM construction.** To test the scalability of the CRT approach, we constructed an elliptic curve using $|D| = 1000000013079299 > 10^{15}$, with $h(D) = 10034174 > 10^7$. This yielded a curve $y^2 = x^3 - 3x + c$ of prime order

n over the prime field $\mathbb{F}_q$, where

$$c = 1222944565023569747153953185348208174607248719445203935546780433368429857904 7;$$
$$q = 28948022309329048855892746252171981646113288548904805961094058424256743169033;$$
$$n = 28948022309329048855892746252171981646453570915825744424557433031688511408013.$$

This curve was obtained by computing the square root of $H_D[A_{71}]$ modulo q, a polynomial of degree $h(D)/2 = 5017087$. The height bound of 21533832 bits was achieved with 438709 small primes p, the largest of which was 53 bits in size. The class polynomial computation took slightly less than a week using 32 cores, approximately 200 days of CPU time. Extracting a root over $\mathbb{F}_q$ took 25 hours of CPU time using NTL [29].

We estimate that the size of $\sqrt{H_D[A_{71}]}$ is over 13 terabytes, and that the size of the Hilbert class polynomial H_D is nearly 2 petabytes. The size of $\sqrt{H_D[A_{71}]} \bmod q$, however, is under 200 megabytes, and less than 800 megabytes of memory (per core) were needed to compute it.

References

[1] Eric Bach. Explicit bounds for primality testing and related problems. *Mathematics of Computation*, 55(191):355–380, 1990.

[2] Juliana Belding, Reinier Bröker, Andreas Enge, and Kristin Lauter. Computing Hilbert class polynomials. In A. J. van der Poorten and A. Stein, editors, *ANTS-VIII*, volume 5011 of *Lect. Notes Comput. Sci.*, pages 282–295. Springer, 2008.

[3] Bruce C. Berndt and Heng Huat Chan. Ramanujan and the modular j-invariant. *Canadian Mathematical Bulletin*, 42(4):427–440, 1999.

[4] Daniel J. Bernstein. Modular exponentiation via the explicit Chinese Remainder Theorem. *Mathematics of Computation*, 76:443–454, 2007.

[5] Gaetan Bisson and Andrew V. Sutherland. Computing the endomorphism ring of an ordinary elliptic curve over a finite field. *Journal of Number Theory*, 2009. To appear, `http://arxiv.org/abs/0902.4670`.

[6] Reinier Bröker. *Constructing elliptic curves of prescribed order.* Proefschrift, Universiteit Leiden, 2006.

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

[8] Reinier Bröker, Kristin Lauter, and Andrew V. Sutherland. Modular polynomials via isogeny volcanoes. Preprint, `http://arxiv.org/abs/1001.0402`, 2009.

[9] Jean-Marc Couveignes and Thierry Henocq. Action of modular correspondences around CM points. In C. Fieker and D. R. Kohel, editors, *ANTS-V*, volume 2369 of *Lect. Notes Comput. Sci.*, pages 234–243. Springer-Verlag, 2002.

[10] Noam D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In D. A. Buell and J. T. Teitelbaum, editors, *Computational Perspectives on Number Theory*, pages 21–76. AMS, 1998.

[11] Andreas Enge. *Courbes algébriques et cryptologie.* Habilitation à diriger des recherches, Université Denis Diderot, Paris 7, 2007.

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

[13] Andreas Enge. Computing modular polynomials in quasi-linear time. *Mathematics of Computation*, 78(267):1809–1824, 2009.

[14] Andreas Enge. `cm`, 0.2 edition, 2010. `http://cm.multiprecision.org/`.

[15] Andreas Enge and François Morain. Generalised Weber functions. I. Technical Report 385608, HAL-INRIA, 2009. `http://hal.inria.fr/inria-00385608`.

[16] Andreas Enge and Reinhard Schertz. Constructing elliptic curves over finite fields using double eta-quotients. *Journal de Théorie des Nombres de Bordeaux*, 16:555–568, 2004.

[17] Andreas Enge and Reinhard Schertz. Modular curves of composite level. *Acta Arithmetica*, 118(2):129–141, 2005.

[18] Andreas Enge and Reinhard Schertz. Singular values of multiple eta-quotients for ramified primes. In preparation, 2010.
[19] Free Software Foundation. *GNU Compiler Collection*, 4.2.4 edition, 2008. `http://gcc.gnu.org/`.
[20] Alice Gee. *Class fields by Shimura reciprocity.* Proefschrift, Universiteit Leiden, 2001.
[21] Alice Gee and Peter Stevenhagen. Generating class fields using Shimura reciprocity. In J. P. Buhler, editor, *ANTS-III*, volume 1423 of *Lect. Notes Comput. Sci.*, pages 441–453, Berlin, 1998. Springer-Verlag.
[22] Torbjörn Granlund et al. `gmp`, 4.3.1 edition, 2009. `http://gmplib.org/`.
[23] Farshid Hajir and Fernando Rodriguez Villegas. Explicit elliptic units, I. *Duke Mathematical Journal*, 90(3):495–521, 1997.
[24] David Harvey. *zn_poly: a library for polynomial arithmetic*, 0.9 edition, 2008. `http://cims.nyu.edu/~harvey/zn_poly`.
[25] David Kohel. Endomorphism rings of elliptic curves over finite fields. PhD thesis, University of California at Berkeley, 1996.
[26] François Morain. Calcul du nombre de points sur une courbe elliptique dans un corps fini: aspects algorithmiques. *Journal de Théorie des Nombres de Bordeaux*, 7(1):111–138, 1995.
[27] François Morain. Advances in the CM method for elliptic curves, 2009. Slides of Fields Cryptography Retrospective Meeting, May 11-15, `http://www.lix.polytechnique.fr/~morain/Exposes/fields09.pdf`.
[28] Reinhard Schertz. Weber's class invariants revisited. *Journal de Théorie des Nombres de Bordeaux*, 14(1):325–343, 2002.
[29] Victor Shoup. *NTL: A library for doing number theory*, 5.5 edition, 2008. `http://www.shoup.net/ntl/`.
[30] Andrew V. Sutherland. Computing Hilbert class polynomials with the Chinese Remainder Theorem. *Mathematics of Computation*, 2010. To appear, `http://arxiv.org/abs/0903.2785`.
[31] Heinrich Weber. *Lehrbuch der Algebra*, volume III. Chelsea, third edition, 1961.

INRIA Bordeaux–Sud-Ouest
E-mail address: `andreas.enge@math.u-bordeaux.fr`

Massachusetts Institute of Technology
E-mail address: `drew@math.mit.edu`