Computation of class polynomials for abelian surfaces

A. Enge\textsuperscript{1}, E. Thomé\textsuperscript{2}

\textsuperscript{1} INRIA/LFANT, Bordeaux ; \textsuperscript{2} INRIA/CARAMEL, Nancy.

Oct. 9th, 2014
Plan

Introduction

CM in genus 1

Genus 2 prerequisites

Algorithm

Computer experiments
Motivations

Algebraic curves over finite fields are nice groups for cryptography.

Desired features: Compact representation of elements.
  Fast arithmetic.
  **Hard** discrete log problem.
  ⇒ Prefer almost prime group order.
Motivations

Algebraic curves over finite fields are nice groups for cryptography.

Desired features:  Compact representation of elements.
  - Fast arithmetic.
  - Hard discrete log problem.
  \[ \Rightarrow \text{Prefer almost prime group order.} \]

Typical candidates.

- Elliptic curves \((g = 1)\)  Studied for crypto for 25+ years.
  - Efficient, secure.

- (Jacobians of) genus 2 curves.
  - Smaller base field for comparable group size.
  - Almost similar efficiency due to recent progress.
  - DL is hard as well.

- Higher genus: DL is comparatively easier. Avoided.
The cardinality issue

**Strategy 1. Direct point counting.**

Pick a curve at random (or select based on arithmetic properties). Compute $\#E(\mathbb{F}_p)$ (or $\#\text{Jac}_C(\mathbb{F}_p)$).

- **Polynomial.** Very fast for small characteristic ($p$-adic).
- **$g = 1$:** fast enough for crypto purposes ($\ell$-adic, SEA).
- **$g = 2$:** now also possible, with some effort (SEA-like).

**Strategy 2**

Select some family of curves for easy point counting. Obtain an instance $(\mathbb{F}_p, E(\mathbb{F}_p), \#E(\mathbb{F}_p))$.

- The **CM method** is such a strategy.
Plan

Introduction

CM in genus 1

Genus 2 prerequisites

Algorithm

Computer experiments
Elliptic curves

- The moduli space of elliptic curves has dimension 1.
- It is parameterised by the $j$-invariant.
- Example: $y^2 = x^3 + ax + b \rightsquigarrow j = 1728 \frac{4a^3}{4a^3 + 27b^2}$.

Endomorphism rings of elliptic curves classified by Deuring.

- In char. 0, either $\mathbb{Z}$ or an order in $\mathbb{Q}(\sqrt{D})$, for some $D < 0$.
- Over finite fields, ordinary: cannot be $\mathbb{Z}$.
  Any ordinary curve over $\mathbb{F}_p$ is the reduction of a curve over $\mathbb{C}$ with same $\text{End}(E)$. 

Computation of class polynomials for abelian surfaces
Strategy

- Pick an imaginary quadratic field $K = \mathbb{Q}(\sqrt{d})$; let $\mathcal{O}_K = \mathbb{Z} + \omega \mathbb{Z}$.
- Let $\mathcal{O} = \mathbb{Z} + f \omega \mathbb{Z}$ be an order in $\mathcal{O}_K$, $\text{disc}(\mathcal{O}) = D = f^2 d$.

Aim at $E \mod$ some (yet unknown) $p$, with $\text{End}(E) = \mathcal{O}$.

- First list $E$ over $\mathbb{C}$ with CM by $\mathcal{O}$.
  - $j$-invariants: roots of Hilbert class polynomial $H_D(x) \in \mathbb{Z}[x]$.
  - Appropriate $p$ have $p = \text{Norm}(\pi \in \mathcal{O}_K$, Weil number).
  - Roots of $H_D \mod p$ are $j$-invariants of their reductions.
  - Those have $\#E(\mathbb{F}_p) = p + 1 \pm \text{Tr} \pi$. 

Effective complex multiplication

Given $D$, what are the curves over $\mathbb{C}$ with CM by $\mathcal{O}$?

Take $a = (\alpha_1, \alpha_2)$ ideal of $\mathcal{O}$ with $\Im(\tau = \frac{\alpha_2}{\alpha_1}) > 0$.

- $\mathbb{C}/a$ has CM by $\mathcal{O}$.
  - $j(a) := j(\tau)$ depends only on the ideal class of $a$.
  - $j$ is a modular function for the action of $SL_2(\mathbb{Z})$ on $\mathcal{H}_1$.
- Curve with invariant $j(a)$ has CM by $\mathcal{O}$,
- There are $h = \# \text{Cl}(\mathcal{O})$ such curves (faithful action).
Main algorithm

- Fix $D < 0$ and Weil number $\pi$.
- Enumerate the $h$ ideal classes of $\mathcal{O}_D$:
  \[
  \left( A_i, \frac{-B_i + \sqrt{D}}{2} \right)
  \]
- Compute over $\mathbb{C}$ the class polynomial
  \[
  H(X) = \prod_{i=1}^{h} \left( X - j \left( \frac{-B_i + \sqrt{D}}{2A_i} \right) \right) \in \mathbb{Z}[X]
  \]
- Find a root $\bar{j}$ modulo $p = \text{Norm} \pi$.
- Curve with that invariant mod $p$ has $\#E = p + 1 \pm \text{Tr} \pi$. 
Complexity

- **Size of $H$**
  - Degree $h \in \tilde{O} \left( \sqrt{|D|} \right)$;
  - Coefficients with $\tilde{O} \left( \sqrt{|D|} \right)$ digits;
  - Total size $\tilde{O} \left( |D| \right)$

- **Evaluation of $j$: $\tilde{O} \left( \sqrt{|D|} \right)$**
  - Precision: $\tilde{O} \left( \sqrt{|D|} \right)$ digits;
  - Multievaluation of the “polynomial” $j$;
  - Arithmetic-geometric mean.

- **Total complexity**
  
  $\tilde{O} \left( |D| \right)$ — quasi-linear in the output size.
Implementation

Record with complex analytic CM (Enge 2009):

- \( D = -2093236031; \)
- \( h = 100000; \)
- Precision 264 727 bits;
- 260 000 seconds = 3 days CPU time;
- 5 GB;
- benefited from using alternative class invariants.

Free, available software, based notably on MPFR/MPC/MPFRCX.
Further algorithms

See Belding–Bröker–Enge–Lauter 2008 and further works for comparison of other methods.

- \(p\)-adic lift.
- Chinese remaindering (CRT):
  - Enumerate CM curves over \(\mathbb{F}_p\), compute \(H \mod p\);
  - Lift to \(\mathbb{Z}\) or directly to \(\mathbb{Z}/P\mathbb{Z}\).
- CRT has the edge for records (Enge–Sutherland 2010):
  - \(D = -1\,000\,000\,013\,079\,299\);
  - \(h = 10\,034\,174\);
  - \(P \approx 2^{254}\);
  - Precision 21\,533\,832 bits;
  - 438\,709 primes of \(\leq 53\) bits;
  - 200 days CPU time;
  - Size mod \(P \approx 200\) MB;
  - Size over \(\mathbb{Z}\) \(\approx 2\) PB (not computed explicitly).
Dupont: One can evaluate $j$ at precision $n$ in time

$$O(M(n) \log n) = \tilde{O}(n).$$

**Idea of the algorithm**

- Newton iterations on a function built with the arithmetic-geometric mean (AGM).
- $j(\tau)$ is a zero of this function.
Genus 1 Theta constants — definition

\[ a, b \in \frac{1}{2} \mathbb{Z}/\mathbb{Z}; \quad q = e^{2\pi i \tau} \]

\[ \theta_{a,b}(\tau) = \sum_{n \in \mathbb{Z}} e^{(2\pi i)(n+a)\tau(n+a)/2+(n+a)b} = e^{2\pi i a b} \sum_{n \in \mathbb{Z}} (e^{2\pi i b})^n q^{(n+a)^2/2} \]

\[ \theta_{0,0}(\tau) = \sum_{n \in \mathbb{Z}} q^{n^2/2} = 1 + 2q^{1/2} + 2q^2 + 2q^{9/2} + \ldots \]

\[ \theta_{0,\frac{1}{2}}(\tau) = \sum_{n \in \mathbb{Z}} (-1)^n q^{n^2/2} = 1 - 2q^{1/2} + 2q^2 - 2q^{9/2} + \ldots \]

\[ \theta_{\frac{1}{2},0}(\tau) = \sum_{n \in \mathbb{Z}} q^{(2n+1)^2/8} = q^{1/8} \left( 1 + 2q + 2q^3 + \ldots \right) \]

\[ \theta_{\frac{1}{2},\frac{1}{2}}(\tau) = 0 \]
Theta constants — duplication formulæ

\[ \theta_{0,0}^2(2\tau) = \frac{\theta_{0,0}^2(\tau) + \theta_{0,1/2}^2(\tau)}{2} \quad \theta_{0,1/2}^2(2\tau) = \sqrt{\theta_{0,0}^2(\tau)\theta_{0,1/2}^2(\tau)} \]
AGM

\[ \theta_{0,0}^2(2\tau) = \frac{\theta_{0,0}^2(\tau) + \theta_{0,\frac{1}{2}}^2(\tau)}{2} \quad \theta_{0,\frac{1}{2}}^2(2\tau) = \sqrt{\theta_{0,0}^2(\tau)\theta_{0,\frac{1}{2}}^2(\tau)} \]

AGM for \( a, b \in \mathbb{C} \)

- \( a_0 = a, \ b_0 = b \)
- \( a_{n+1} = \frac{a_n + b_n}{2} \)
- \( b_{n+1} = \sqrt{a_n b_n} \), closer to \( a_{n+1} \) than to its opposite.
- **converges quadratically** towards a common limit \( \text{AGM}(a, b) \)

Evaluated in time \( O(M(n) \log n) \) at precision \( n \).

For \( \tau \in \text{some region of } \mathcal{H}_1 \),

\[ \left\{ \left( \theta_{0,0}^2, \theta_{0,\frac{1}{2}}^2 \right)(2^n \tau) \right\} \]

is the AGM sequence starting from \( \tau \) (whence the limit is 1).
Theta quotients

The AGM is an homogeneous bivariate function on $\mathbb{C}$. We define:

$$\text{AGM}(a, b) = a \cdot \text{AGM}(1, b/a) =: a \cdot M(b/a)$$

- $k'(\tau) = \left(\frac{\theta_{0, \frac{1}{2}}(\tau)}{\theta_{0, 0}(\tau)}\right)^2$
- $k(\tau) = \left(\frac{\theta_{\frac{1}{2}, 0}(\tau)}{\theta_{0, 0}(\tau)}\right)^2$
- $k^2(\tau) + k'^2(\tau) = 1$
- $j = 256\frac{(1-k'^2+k'^4)^3}{k'^4(1-k'^2)^2}$

$j$ can be computed from $k'$
Newton iterations

\[ M(k'(\tau)) = \frac{1}{\theta_{0,0}^2(\tau)}, \]
\[ M(k(\tau)) = M(k'(-1/\tau)) = \frac{1}{\theta_{0,0}^2(-1/\tau)} = \frac{i}{\tau \theta_{0,0}^2(\tau)}, \]
\[ k^2(\tau) + k'^2(\tau) = 1 \]
\[ f_\tau(x) = iM(x) - \tau M(\sqrt{1 - x^2}) \]
\[ f_\tau(k'(\tau)) = 0 \]

\[ x_{n+1} \leftarrow x_n - \frac{f_\tau(x_n)}{f'_\tau(x_n)} \]

converges quadratically towards \( k'(\tau) \)

Evaluated in time \( O(M(n) \log n) \) at precision \( n \).

Caution

Care must be taken to consider \( \tau \) for which the homogeneous AGM converges to 1 (which gives \( M(k'(\tau)) = \frac{1}{\theta_{0,0}^2(\tau)} \)).
Plan

Introduction

CM in genus 1

Genus 2 prerequisites

Algorithm

Computer experiments
Generalization: Genus 2 CM

Let \( K \) be a CM field.

\[
\begin{align*}
K & \quad 2, \text{ totally imaginary} \\
K_0 & \quad g = 2, \text{ totally real} \\
\mathbb{Q} & 
\end{align*}
\]

Workplan

- Enumerate principally polarized abelian varieties (PPAVs) with complex multiplication by \( \mathcal{O}_K \) (\( \text{End} = \mathcal{O}_K \)).
- Compute their invariants in \( \mathbb{C} \) (\( g = 2 \), three invariants).
- Compute their defining polynomials: Igusa class polynomials.
- Recognize these (triples of) polynomials in \( \mathbb{Q}[x] \).

The larger the discriminants, the bigger the polynomials.
Various approaches

- Complex analytic method: Spallek, Weng, Streng.
- $p$-adic: Gaudry, Houtmann, Kohel, Ritzenthaler, Weng, Carls, Lubicz.
- CRT: Eisentrager, Lauter, Bröker, Gruenewald, Robert.

Focus on the **complex analytic method**

- Streng: complete algorithm, and complexity upper bounds.
- Improve on keypoint: *computation of invariants* analytically.
- Recognize irreducible factors of class polynomials.
Preferred defining equation for $K$: $x^4 + Ax^2 + B$, with $A^2 - 4B = \Box \times \text{disc}(K_0)$.

Let $D = \text{disc}(K_0)$, and $A$ minimal $\Rightarrow$ invariants $[D, A, B]$.

The CM field $K$ may be either:

- Galois with $\text{Gal}(K/\mathbb{Q}) = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$; (degenerates to $g = 1$).
- Galois with $\text{Gal}(K/\mathbb{Q}) = \mathbb{Z}/4\mathbb{Z}$; (cyclic case, rare).
- non-Galois, with $\text{Gal}(L/\mathbb{Q}) = D_4 = \mathbb{Z}/4\mathbb{Z} \rtimes \mathbb{Z}/2\mathbb{Z}$; (typical).

Study of the Galois structure reveals:

- two non-conjugate pairs of embeddings $K \hookrightarrow \mathbb{C}$;
- the reflex field $K^r$ of $K$, which is another CM field.
(2/5): Period matrices

Siegel upper-half space $\mathcal{H}_2$: symm. + pos. def. imag. part.

- $\text{Sp}_4(\mathbb{Z})$ acts on $\mathcal{H}_2$: $\begin{pmatrix} A & B \\ C & D \end{pmatrix}.\tau = (A\tau + B)(C\tau + D)^{-1}$.

- $\mathcal{F}_2$: fundamental domain for $\text{Sp}_4 \backslash \mathcal{H}_2$.

$\text{PPAV}$ = $\mathbb{Z}$-lattice in $\mathbb{C}^2$ + Riemann form $\rightarrow$ period matrix $\tau = \begin{pmatrix} \tau_1 & \tau_3 \\ \tau_3 & \tau_2 \end{pmatrix} \in \mathcal{H}_2$. 

Computation of class polynomials for abelian surfaces
Theta constants for \( \mathbf{a} = (a_1, a_2), \mathbf{b} = (b_1, b_2), a_i, b_i \in \{0, 1/2\} \):

\[
\theta_{[\mathbf{a}, \mathbf{b}]}(\tau) = \sum_{\mathbf{n} \in \mathbb{Z}^2} \exp \left( i\pi \left[ (\mathbf{n} + \mathbf{a})\tau (\mathbf{n} + \mathbf{a})^t + 2(\mathbf{n} + \mathbf{a})\mathbf{b}^t \right] \right).
\]

- Numbering (Dupont) \( \theta_{[\mathbf{a}, \mathbf{b}]} = \theta_{2b_1+4b_2+8a_1+16a_2} \).
- 10 even theta constants: \( \theta_{0,1,2,3,4,6,8,9,12,15} \), other are 0.

**Theta constants** are used to compute *invariants*.

### Duplication formulae

We have unambiguous formulae:

\[
4\text{-uple} \left( \theta_{0,1,2,3}(\tau/2) \right) \rightarrow 10\text{-uple} \left( \theta_{0,1,2,3,4,6,8,9,12,15}^2(\tau) \right).
\]
The moduli space of 2-dimensional PPAVs has dimension 3. **Igusa invariants** can be computed from $\theta_{0,1,2,3,4,6,8,9,12,15}$.

- Several invariant sets floating around.
- Some “smaller” than others.
- Define $(i_1, i_2, i_3)$ as those proposed by Streng.

\begin{align*}
i_1 &= \frac{l_4(l_2 l_4 - 3 l_6)}{2 l_{10}} \quad i_2 = \frac{l_2 l_4^2}{l_{10}} \quad i_3 = \frac{l_4^5}{l_{10}^2}.
\end{align*}
Consider $S(K)$ the set of PPAVs with CM by $\mathcal{O}_K$.
The set $\{i_1(\tau), \tau \in S(K)\}$ is defined over $\mathbb{Q}$.

- Minimal polynomials $H_1, H_2, H_3$ in $\mathbb{Q}[x]$.
- Better: $\{i_{1,2,3}(\tau)\}$ a 0-dimensional set in $\mathbb{C}^3$, defined over $\mathbb{Q}$.
- Triangular (Hecke) representation: $H_1$, $\hat{H}_2$, $\hat{H}_3$, with:

$$\hat{H}_2(i_1) = H_1'(i_1)i_2.$$ 

The triple $(H_1, \hat{H}_2, \hat{H}_3)$ is our target.

Obstacles:

- Large degree, (very) large coefficients.
- Need large precision for complex invariants, so that rational polynomials may be recognized.
Plan

Introduction

CM in genus 1

Genus 2 prerequisites

Algorithm

Computer experiments

Computation of class polynomials for abelian surfaces
Workplan (again)

1. List period matrices
2. Compute $\theta$-constants
3. Compute class polynomials
4. Compute a curve example
Plan

Algorithm

Principally polarized abelian varieties with CM by $\mathcal{O}_K$
Computing complex invariants
From $\theta$-constants to class polynomials
Let $\mathfrak{a}$ be an $\mathcal{O}_K$-ideal with:

- $(\mathfrak{a}\overline{\mathfrak{a}}\mathcal{D}_K/\mathbb{Q})^{-1} = (\xi)$,
- $\Phi(\xi) \in i\mathbb{R}^+*$ for some CM-type $\Phi$.

Such $\mathfrak{a}$'s yield period matrices $\Omega \in \mathcal{M}_2(K^r) \hookrightarrow \mathcal{H}_2 \twoheadrightarrow \mathcal{F}_2$.

Conversely, all PPAVs with CM by $\mathcal{O}_K$ are obtained this way.

Easy plan: enumerate representatives of $\text{Cl}(\mathcal{O}_K)$ to find both.

Way more satisfactory: enumerate only irreducible components, working with Shimura group $\mathcal{C}(K)$ and the reflex typenorm map.
Plan

Algorithm

Principally polarized abelian varieties with CM by $\mathcal{O}_K$

Computing complex invariants

From $\theta$-constants to class polynomials
Computing theta constants

Input: $\tau \in \mathcal{F}_2$, whose entries are algebraic numbers (in $K^r$).

Goal: theta constants $\theta_{0,1,2,3,4,6,8,9,12,15}$ (and later $i_{1,2,3}$).

Large precision $N$ needed to successful reconstruct $H_1, \hat{H}_2, \hat{H}_3$.

Upper bounds on $N$ exist. Difficult to make it tight.

Two strategies for computing $\theta$'s from $\tau$.

- $q$-expansion of $\theta_{0,1,2,3}(\tau/2)$, letting $q_k = \exp(i\pi\tau_k/2)$:

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

  Summation over $O(N)$ terms, can be done in $O(NM(N))$.
  Finish with duplication formulae.

- Faster: Newton lifting.
Dupont defines a **Borchardt sequence** as \((x_n, y_n, z_n, t_n) \in \mathbb{C}^4\):

\[
\begin{align*}
    x_{n+1} &= \frac{1}{4}(x_n + y_n + z_n + t_n), \\
    y_{n+1} &= \frac{1}{2}(\sqrt{x_n}\sqrt{y_n} + \sqrt{z_n}\sqrt{t_n}), \\
    z_{n+1} &= \frac{1}{2}(\sqrt{x_n}\sqrt{z_n} + \sqrt{y_n}\sqrt{t_n}), \\
    t_{n+1} &= \frac{1}{2}(\sqrt{x_n}\sqrt{t_n} + \sqrt{y_n}\sqrt{z_n}).
\end{align*}
\]

- **Choice of** \(\sqrt{\cdot}\) **at each iteration.**
- **Starting** \((x_0, y_0, z_0, t_0)\): set of possible limits \(B_2(x_0, y_0, z_0, t_0)\).
- **Forcing** **consistent** choice of roots: \(B_2(x, y, z, t)\) **well defined.**

Let \(\mathcal{U} = \{\tau \in \mathcal{H}_2, \ B_2(\theta_{0,1,2,3}^2(\tau)) = 1\}\). At least \(\mathcal{F}_2 \subset \mathcal{U}\).

**Homogeneity**

\[
B_2(\lambda x, \lambda y, \lambda z, \lambda t) = \lambda B_2(x, y, z, t).
\]
Exploiting action of $\text{Sp}_4(\mathbb{Z})$

Action of $\Gamma_2$ on the theta constants

Let $\tau \in \mathcal{H}_2$. Then

\[
\begin{align*}
\left( \theta_j^2( (\mathcal{J}\mathcal{M}_1)^2 \tau) \right)_{j=0,1,2,3} &= -i\tau_1 \left( \theta_j^2(\tau) \right)_{j=4,0,6,2}, \\
\left( \theta_j^2( (\mathcal{J}\mathcal{M}_2)^2 \tau) \right)_{j=0,1,2,3} &= -i\tau_2 \left( \theta_j^2(\tau) \right)_{j=8,9,0,1}, \\
\left( \theta_j^2( (\mathcal{J}\mathcal{M}_3)^2 \tau) \right)_{j=0,1,2,3} &= (\tau_3^2 - \tau_1\tau_2) \left( \theta_j^2(\tau) \right)_{j=0,8,4,12}.
\end{align*}
\]

Important: if $(\mathcal{J}\mathcal{M}_1)^2.\tau \in \mathcal{U}$, then $B_2(\theta_{4,0,6,2}^2(\tau)) = \frac{1}{-i\tau_1}$.

Conjecture

For $\tau \in \mathcal{F}_2$, $i \in \{0, 1, 2\}$: $(\mathcal{J}\mathcal{M}_i)^2.(\tau) \in \mathcal{U}$.
\( \theta_{0,1,2,3}(\tau/2) \) as solutions of an equation

Input: \( \tau \in \mathcal{F}_2 \) known (to any precision we like).
Initially: low-precision \( \theta_{0,1,2,3}(\tau/2) \).

- Use duplication formulae to deduce \( \theta_{0,1,2,3,4,6,8,9,12,15}^2(\tau) \).
- Use \( B_2 \) computations to deduce coefficients of \( \tau \).
- The accurate \( x_{0,1,2,3} = \theta_{0,1,2,3}(\tau/2) \) are solutions to

\[
\text{complicated-} B_2 \text{-calculation}(x_{0,1,2,3}) = \tau.
\]

Newton: use this feedback loop to find \( \theta_{0,1,2,3}(\tau/2) \).

- Keeping track of derivatives is messy.
- A secant method also works, and is actually more convenient.
Computation of $\theta_{0,1,2,3}^2$ by Newton lifting

Convergence of the Newton iteration is quadratic:
- each iteration (almost) doubles the precision.
- it is possible to “lift higher” without restarting from scratch.

Complexity of the algorithm: quasi-linear $O(M(N) \log N)$. 
Performance measurements

\[ \tau = \left( \frac{-1+5i}{2}, \frac{i}{6}, \frac{-1+7i}{2} \right) \]
\[ \tau = \left( \frac{2+10i}{7}, \frac{1+2i}{6}, \frac{1+2i}{10} + 8i \right) \]

<table>
<thead>
<tr>
<th>bits</th>
<th>( \text{MAGMA} )</th>
<th>( \text{cmh-naive} )</th>
<th>( \text{cmh-Newton} )</th>
<th>( \text{MAGMA} )</th>
<th>( \text{cmh-naive} )</th>
<th>( \text{cmh-Newton} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 2^{11} )</td>
<td>0.46</td>
<td>0.02</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>( 2^{12} )</td>
<td>3.4</td>
<td>0.04</td>
<td>0.17</td>
<td>0.04</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>( 2^{13} )</td>
<td>26</td>
<td>0.08</td>
<td>1.1</td>
<td>0.20</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td>( 2^{14} )</td>
<td>210</td>
<td>0.24</td>
<td>8.2</td>
<td>1.0</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>( 2^{15} )</td>
<td>1700</td>
<td>0.69</td>
<td>60</td>
<td>5.2</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>( 2^{16} )</td>
<td>6.4</td>
<td>2.0</td>
<td>430</td>
<td>27</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>( 2^{17} )</td>
<td>32</td>
<td>5.7</td>
<td>3100</td>
<td>130</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>( 2^{18} )</td>
<td>160</td>
<td>16</td>
<td>720</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{19} )</td>
<td>770</td>
<td>39</td>
<td>3100</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{20} )</td>
<td>3200</td>
<td>98</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{21} )</td>
<td></td>
<td>240</td>
<td>230</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{22} )</td>
<td></td>
<td>560</td>
<td>530</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{23} )</td>
<td></td>
<td>1400</td>
<td>1300</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{24} )</td>
<td></td>
<td>3200</td>
<td>3000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{25} )</td>
<td></td>
<td>7600</td>
<td>7100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{26} )</td>
<td></td>
<td>16000</td>
<td>16000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table:

Computation of \( \theta_0(\tau) \) (Intel i5-2500, 3.3GHz; \text{MAGMA}-2.19.4; \text{cmh}-1.0)
Plan

Algorithm

Principally polarized abelian varieties with CM by $\mathcal{O}_K$
Computing complex invariants
From $\theta$-constants to class polynomials
Reconstruction

\( \theta \)-constants \( \rightsquigarrow \) three Igusa invariants : trivial.

From these, we compute:

- **Product trees** yield \( H_1, \hat{H}_2, \hat{H}_3 \in \mathbb{R}[x] \).
- Their coefficients belong to the quadratic real \( K_{r}^\prime \).

Recognize \( x \in \mathbb{R} \) as short vector in:

\[
\begin{pmatrix}
1 & \kappa_1 & 0 & 0 \\
\sqrt{D'} & 0 & \kappa_2 & 0 \\
x & 0 & 0 & \kappa_3
\end{pmatrix}
\]

Success criterion: **smooth denominators**.

- Denominators can be predicted to some extent (not done).
- As long as reconstruction fails, keep on lifting \( \theta_{0,1,2,3}^2(\tau) \).
  At most we’re lifting twice higher than what we would need if
  we had sharp bounds on denominators.
Plan

Introduction
CM in genus 1
Genus 2 prerequisites
Algorithm
Computer experiments
Implementation

- Number theoretic computations: \( \mathcal{C}(K) \), (reduced) period matrices
  - Pari/GP
  - negligible effort
- Evaluation of theta and invariants
  - C
  - Libraries: GMP, MPFR, MPC
  - MPI for parallelisation
- Polynomial operations
  - MPFRGCX
  - MPI for (partial) parallelisation

Computation of class polynomials for abelian surfaces
Software

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

- GPLv3+
- ./configure --with-gmp=... ... --enable-mpi
  make install
- Period matrices: cmh-classpol.sh -p 35 65
- Class polynomials: cmh-classpol.sh -f 35 65
- Curve for checking: cmh-classpol.sh -c 35 65
- Using MPI:
  mpirun -n 4 cm2-mpi -i 965_35_65.in -o H123.pol
## Two baby examples

\[
X^4 + 144X^2 + 3500 = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/30\mathbb{Z}
\]

<table>
<thead>
<tr>
<th>Step</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>0.2</td>
</tr>
<tr>
<td>Base, 2000 bits</td>
<td>0.6</td>
</tr>
<tr>
<td>Lift, 3984 bits</td>
<td>0.8</td>
</tr>
<tr>
<td>Lift, 7944 bits</td>
<td>2.1</td>
</tr>
<tr>
<td>Reconstruction</td>
<td>0.1</td>
</tr>
<tr>
<td>Lift, 15846 bits</td>
<td>6.2</td>
</tr>
</tbody>
</table>

\[
H_1, \hat{H}_2, \hat{H}_3 \in \mathbb{C}[X]
\]

\[
H_1, \hat{H}_2, \hat{H}_3 \in K_0^r[X]
\]

<table>
<thead>
<tr>
<th>Step</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check</td>
<td>0.8</td>
</tr>
<tr>
<td>Total (incl. I/O)</td>
<td>12.4</td>
</tr>
</tbody>
</table>

\[
X^4 + 134X^2 + 712 = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/60\mathbb{Z}
\]

<table>
<thead>
<tr>
<th>Step</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td>0.3</td>
</tr>
<tr>
<td>Base, 2000 bits</td>
<td>1.1</td>
</tr>
<tr>
<td>Lift, 3988 bits</td>
<td>1.6</td>
</tr>
<tr>
<td>Lift, 7958 bits</td>
<td>4.4</td>
</tr>
<tr>
<td>Lift, 15886 bits</td>
<td>13.1</td>
</tr>
<tr>
<td>Reconstruction</td>
<td>0.2</td>
</tr>
<tr>
<td>Lift, 31744 bits</td>
<td>38.7</td>
</tr>
</tbody>
</table>

\[
H_1, \hat{H}_2, \hat{H}_3 \in \mathbb{C}[X]
\]

\[
H_1, \hat{H}_2, \hat{H}_3 \in K_0^r[X]
\]

<table>
<thead>
<tr>
<th>Step</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check</td>
<td>0.7</td>
</tr>
<tr>
<td>Total (incl. I/O)</td>
<td>69.2</td>
</tr>
</tbody>
</table>

Timings in seconds for two examples (Intel i5-2500, 3.3GHz).
One jumbo experiment

How far can we go?

- \[ K = \mathbb{Q}[X]/(X^4 + 1357X^2 + 3299), \quad K_0 = \mathbb{Q}(\sqrt{1828253}). \]
- \( \mathfrak{c} \simeq \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/5004\mathbb{Z}; \quad \mathfrak{c} = 20016. \)

Computation breakdown:

- 10 008 symbolic period matrices: \text{minutes}.
- Lift up to 2 000 000 bits: \text{hours} (640 cores).
- Lift up to 8 000 000 bits: \text{3 days} (160 cores, more RAM).
- Computing polynomials: \text{3 days} (24 cores).
- Recognizing coefficients: \text{2 days} (480 cores).
- Disk size for class polynomial triple: \text{90 GB}.

\( \text{lc}(H_1) \) has 8 884 distinct prime factors, largest is 1 506 803 839.
A curve

\[
\pi = 2587584949432298\alpha^3 + 598749326588980\alpha^2 +
3489110163205995872\alpha - 17626367557116479015,
\]
\[
p^2 = \text{Norm}(\pi) = (2^{128} + 5399685)^2,
\]
\[
y^2 = 329105434147215182703081697774190891717x^5 +
219357712933218699650940059644263138156x^4 +
94773520721686083389380651745963315116x^3 +
13612280714446818104030347122109215819x^2 +
224591198286067822213326173663420732292x +
62350272396394045327709463978232206155,
\]
\[
\chi = t^4 - s_1 t^3 + s_2 t^2 - ps_1 t + p^2, \quad (s_1 = -72130475900828407780,
\]
\[
s_2 = 1980610692179048658315492237655054733182),
\]
\[
\#J = (p^2 + 1) - (p + 1)s_1 + s_2 = 2^4 \cdot 3433 \cdot p_{73}.
\]
Conclusion

- Complex analytic CM construction is effective in genus 2, not just for ridiculously small examples;
- We don’t meet the sky-large class number requirements though;
- Computing $\theta$-constants is fast. Never say it’s a bottleneck. There’s available software!

Further improvements:

- Higher genus?
- Prove the conjectures? (note: there are trivial workarounds anyway).
- Improve on our recognition step, which is too slow.
- Compute $\theta(\tau, z)$, not just $\theta(\tau, 0)$.
- Improve the CRT method to make it as effective.