Hilbert class polynomials and modular polynomials

Hamish Ivey-Law
hamish.ivey-law@inria.fr

1LFANT team, INRIA Bordeaux Sud-Ouest

2Institut de Mathématiques de Bordeaux
Université de Bordeaux 1

14th of January, 2015
This talk is about
- Hilbert class polynomials: \texttt{polclass}
- modular polynomials: \texttt{polmodular}

For each of these topics we will
- Briefly recall the main definitions and context.
- Describe (in broad strokes) the algorithm(s) to compute them.
- Describe (and solicit suggestions for) the PARI/GP interface to the implementation.

The algorithms for computing Hilbert class polynomials and modular polynomials are due to Andrew Sutherland and his collaborators (including G. Bisson, R. Bröker, A. Enge, K. Lauter).
What is $H_D(X)$?

- Let $D \leq -3$ be a quadratic discriminant and denote the order of discriminant $D$ by $\mathcal{O}_D$.
- The $j$-invariant of the elliptic curve $\mathbb{C}/\mathcal{O}_D$ is an algebraic integer whose minimal polynomial $H_D(X)$ is the **Hilbert class polynomial** for the discriminant $D$.
- The degree $h(D)$ of $H_D(X)$ is the **class number** of $D$.
- The **norm equation** for $D$ is

$$4p = t^2 - v^2 D$$

for some integers $p, t$ and $v$, where $p$ is prime.
- $H_D(X)$ splits completely over $\mathbb{F}_p$ if $p$ satisfies the norm equation.
How big is $H_D(X)$?

- Total size of $H_D(X)$ is $O(|D|^{1+\varepsilon})$ bits.
  - Degree is $O(|D|^{1/2} \log |D|)$
  - Let $B$ be an upper bound for the height of the coefficients. Then $\log(B)$ is $O(|D|^{1/2} \log^2 |D|)$

<table>
<thead>
<tr>
<th>$D$</th>
<th>$h(D)$</th>
<th>$h(D) \log(B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6 + 3$</td>
<td>105</td>
<td>113KB</td>
</tr>
<tr>
<td>$10^8 + 3$</td>
<td>1702</td>
<td>33MB</td>
</tr>
<tr>
<td>$10^{10} + 3$</td>
<td>10538</td>
<td>2GB</td>
</tr>
<tr>
<td>$10^{12} + 3$</td>
<td>124568</td>
<td>265GB</td>
</tr>
<tr>
<td>$10^{14} + 3$</td>
<td>1425472</td>
<td>39TB</td>
</tr>
</tbody>
</table>
When $p$ satisfies the norm equation, $H_D(X)$ splits completely over $\mathbb{F}_p$ and its roots are the $j$-invariants of the elliptic curves whose endomorphism rings are isomorphic to $\mathcal{O}_D$.

This allows us to compute $H_D(X)$ modulo such a $p$. Suppose $4p = t^2 - v^2 D$ for some integers $t$ and $v$. Then

1. Search for a curve $E/\mathbb{F}_p$ whose (absolute) trace is $t$.
2. Search for a curve $E'/\mathbb{F}_p$ which is isogenous to $E$ and has endomorphism ring $\mathcal{O}_D$. Its $j$-invariant $j_0$ gives a root of $H_D(X) \pmod{p}$.
3. Enumerate all curves with endomorphism ring $\mathcal{O}_D$ using the action of $\text{cl}(D)$, starting from $j_0$.
4. Compute $H_D \pmod{p}$ as $H_D(X) = \prod_{\text{End}(j) = \mathcal{O}_D} (X - j)$. 
The complete algorithm to compute $H_D(X)$ just applies the CRT:

1. Select a set $S$ of split primes such that $\prod_{p \in S} p > 4B$.
2. Compute a suitable presentation for $\text{cl}(D)$.
3. For each $p \in S$,
   1. Compute $H_D(X) \pmod{p}$ (uses the presentation of $\text{cl}(D)$).
   2. Update CRT for each coefficient of $H_D(X) \pmod{p}$.
4. Deduce the coefficients of $H_D(X)$.

To compute $H_D(X)$ over $\mathbb{Z}/M\mathbb{Z}$ one still has to compute $H_D \pmod{p}$ for sufficiently many primes $p$ to determine $H_D$ over $\mathbb{Z}$, even when $M$ is small. Using the “explicit CRT” allows us to reduce the space required, but not the overall running time.

Interface: `polclass(D, {x = ’x})`
Assuming the GRH, to calculate $H_D(X)$ modulo an integer $M$, the algorithm

- uses $O(|D|^{1/2+\varepsilon} \log(M))$ space, and
- has expected running time $O(|D|^{1+\varepsilon})$. 
Minimal polycyclic presentations
  - Small generators, not a basis

Isogeny volcanoes
  - depth
  - navigation up/down
  - find level
  - path to surface/floor

Affine models for modular curves $X_1(N)$ for $N \leq 50$.

Find $j$-invariant of curve with given trace.

Find $j$-invariant with given endomorphism ring

Test for supersingularity (over arbitrary finite base field).
Example

```gp
D = -133563
%1 = -133563

coredisc(D, 1)
%2 = [-3, 211]

quadclassunit(D)
%3 = [70, [70], [Qfb(91, 47, 373)], 1]

H = polclass(D);
time = 1,691 ms.
```
What is $\Phi_\ell(X, Y)$?

- The *modular polynomial* of level $\ell$ parameterises $\ell$-isogenous pairs of elliptic curves over $\mathbb{C}$:
  
  $$\Phi_\ell(j(E_1), j(E_2)) \text{ if and only if } E_1 \text{ and } E_2 \text{ are } \ell\text{-isogenous.}$$

- This interpretation remains valid over any field of characteristic not dividing $\ell$. 
How big is $\Phi_\ell(X, Y)$?

- Total size of $\Phi_\ell(X, Y)$ is $O(\ell^{3+\varepsilon})$ bits.
  - Degree in each variable is $\ell + 1$.
  - Let $B$ be an upper bound for the height of the coefficients. Then $\log(B)$ is $6\ell \log(\ell) + O(\ell)$.

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>size (MB)$^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>2.65</td>
</tr>
<tr>
<td>211</td>
<td>27.6</td>
</tr>
<tr>
<td>307</td>
<td>90.5</td>
</tr>
<tr>
<td>1009</td>
<td>3857.0</td>
</tr>
</tbody>
</table>

$^1$N.B. This is half what we quote later because $\Phi_\ell$ is symmetric, a fact not easily exploited in Pari.
Setup for odd level \( \ell \). Let

- \( \mathcal{O} \) be an imaginary quadratic order of discriminant \( D \) whose class number satisfies \( h(D) \geq \ell + 2 \),

- \( p \equiv 1 \pmod{\ell} \) be a prime satisfying \( 4p = t^2 - \nu^2 \ell^2 D \) for some integers \( t \) and \( \nu \) with \( \ell \nmid \nu \), and

- \( R = \mathbb{Z} + \ell \mathcal{O} \) be the order of index \( \ell \) in \( \mathcal{O} \).

Such \( D \) and \( p \) are easy to find.
Modular polynomial modulo a (small) split prime

With the setup on the previous slide, $\Phi_\ell(X, Y) \pmod{p}$ is computed as follows:

1. Find a root of $H_\Theta$ over $\mathbb{F}_p$.
2. Enumerate the roots $j_i$ of $H_\Theta$ and identify $\ell$-isogeny cycles.
3. For each $j_i$ find an $\ell$-isogenous $j$-invariant $j'_i$ on the floor of the $\ell$-volcano.
4. Enumerate the roots of $H_R$ and identify $\ell^2$-isogeny cycles.
5. For each $j_i$ compute $\Phi_\ell(X, j_i) = \prod (X - j_k)$ where the product is over the neighbours of $j_i$ in its $\ell$-isogeny cycle together with the $\ell^2$-isogeny cycle containing $j'_i$.
6. Interpolate $\Phi_\ell \in (\mathbb{F}_p[Y])[X]$ using the $j_i$ and the polynomials $\Phi_\ell(X, j_i)$. 
Given an odd prime $\ell$,

1. Find a suitable order $\mathcal{O}$ of discriminant $D$ where $h(D) \geq \ell + 2$.

2. Compute the class polynomial $H_{\mathcal{O}}$ over $\mathbb{Z}$.

3. Select a sufficiently large set $S$ of primes of the form $4p = t^2 - \ell^2 v^2 D$ where $\ell \nmid v$, $p \equiv 1 \pmod{\ell}$.

4. For each prime $p$ in $S$,
   - Compute $\Phi_{\ell}(X, Y) \pmod{p}$ using the previous algorithm using $\mathcal{O}$ and $H_{\mathcal{O}}$.
   - Update CRT data using $\Phi_{\ell} \pmod{p}$.

5. Finalise CRT computation and output $\Phi_{\ell}$ in $\mathbb{Z}[X, Y]$. 
Assuming the GRH, to calculate $\Phi_\ell(X, Y)$ modulo an integer $M$, the algorithm

- uses $O(\ell^2 (\log \ell)^2 + \ell^2 \log M)$ space, and
- has expected running time $O(\ell^3 (\log \ell)^3 \log \log \ell)$. 
Example

Interface: `polmodular(L, {x = 'x}, {y = 'y}, {compute_derivs = 0})`

gp> polmodular(101); \ about 5.5MB
   *** polmodular: Warning: increasing stack size to 32000000.
time = 6,174 ms.
gp> polmodular(199); \ about 47MB
   *** polmodular: Warning: increasing stack size to 32000000.
   *** polmodular: Warning: increasing stack size to 64000000.
   *** polmodular: Warning: increasing stack size to 128000000.
   *** polmodular: Warning: increasing stack size to 256000000.
time = 57,387 ms.
gp> polmodular(199, random(Mod(1, 12)), 'x); \ about 16kB
   *** polmodular: Warning: increasing stack size to 16000000.
   *** polmodular: Warning: increasing stack size to 32000000.
   *** polmodular: Warning: increasing stack size to 64000000.
time = 51,637 ms.
Bill A. has tested `polmodular` on a machine with 96 cores, and lots of RAM.

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>result size (GB)</th>
<th>stack size (GB)</th>
<th>wall clock time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1009</td>
<td>7.74</td>
<td>32</td>
<td>2m38s</td>
</tr>
<tr>
<td>2003</td>
<td>66.3</td>
<td>256</td>
<td>26m14s</td>
</tr>
<tr>
<td>3001</td>
<td>234.0</td>
<td>1000</td>
<td>2h16m29s</td>
</tr>
</tbody>
</table>
Summary of new features

- Hilbert class polynomials
  - modulo $M$ or over $\mathbb{Z}$
  - with various modular functions (⋆)
- Modular polynomials
  - modulo $M$ or over $\mathbb{Z}$
  - pre-instantiated
  - non-prime level (⋆)
  - with various modular functions (⋆)
- Navigating isogeny volcanoes
  - Depth, find level
  - Move up/down, path to surface/floor
  - Enumerate surface
  - Produce partial/complete (labelled) graph (?)

- Minimal polycyclic presentations
- Testing supersingularity
- Optimised equations for $X_1(N)$ for $N \leq 50$
- Find curves with given trace
- Find curve with given endo ring
- Explicit CRT (⋆)
- Calculate endomorphism ring of a given curve
- Action of $\text{cl}(\mathcal{O})$ on $\text{Ell}_{\mathcal{O}}(\mathbb{F}_p)$
- Enumerate kernel of $\text{cl}(\mathbb{Z} + N\mathcal{O}) \rightarrow \text{cl}(\mathcal{O})$

(⋆): something planned but not yet finished; (?) : something that could be done if you want. Send suggestions to hamish.ivey-law@inria.fr!