Computing Heegner Points in Pari/gp

John Cremona
University of Nottingham, UK

14 September, 2004

What are Heegner points?

They are rational points defined on certain special elliptic curves

In this talk, they will be rational points of infinite order defined only on elliptic curves

- *defined over* $\mathbb{Q}$ and
- of *Mordell-Weil rank* 1.

We will say little about their significant theoretical importance (in almost proving the BSD conjectures for curves of rank 1), but only discuss how they may be used to compute explicit nontrivial rational points on rank one elliptic curves.

Plan of the talk

- A first (easy) example
- Some theory, leading to
- a recipe
- Some tricks
- Implementations
- Bigger examples

A first example

The curve $5160J_1$:

$$y^2 = x^3 - x^2 + 399549679x + 2496643493445$$

has rank 1; its generator has canonical height (predicted by the BSD conjecture) approximately 13.07.

A first example

The curve $5160J_1$:

$$y^2 = x^3 - x^2 + 399549679x + 2496643493445$$

has rank 1; its generator has canonical height (predicted by the BSD conjecture) approximately 13.07.

The generator is

$$P = \left(\frac{770528077163}{6195121}, \frac{6854768827281328 50}{15419656169}\right)$$
$$= \left(\frac{770528077163}{2489^2}, \frac{685476882728132850}{2489^3}\right).$$

A simple gp script

```
e=ellinit( [0,-1,0,399549679,2496643493445] );
w1=e.omega[1]
n=ellglobalred(e)[1];
print("N = ",n);
nan=40000;
an=ellan(e,nan);
b=6677; d=-71; h=7; h0=4; ai=[1,2,3,4,5,8,10]
tau=(-b+sqrt(d))/(2*n)
qi=vector(h0,k,exp(2*Pi*I*tau/ai[k]))
xi=vector(h0,k,1)
si=vector(h0,k,0)
for(j=1,nan,cn=an[j]/j; \
for(k=1,h0,xi[k]*=qi[k]; si[k]+=cn*xi[k]));
s=real(si[1]+2*(si[2]+si[3]+si[4]));
z=(2*s+3*w1)/32
p=ellztopoint(e,z)
xp=bestappr(real(p[1]),10^10)
yp=ellordinate(e,xp)[1]
p=[xp,yp]
if(ellisoncurve(e,p),print("P = ",p,"\nHeight = ",ellheight(e,p)))
```

A little theory

- $E/\mathbb{Q}$ is modular;

A little theory

- $E/\mathbb{Q}$ is modular;
- so there is a map $\varphi\colon X_0(N) \to E$ where $N = \mathrm{cond}(E)$ and $X_0(N)$ is the modular curve (defined over $\mathbb{Q}$);

A little theory

- $E/\mathbb{Q}$ is modular;
- so there is a map $\varphi\colon X_0(N) \to E$ where $N = \mathrm{cond}(E)$ and $X_0(N)$ is the modular curve (defined over $\mathbb{Q}$);
- so we might hope that carefully chosen points $\tau \in X_0(N)$ might map to rational points on $E \ldots$

A little theory

- $E/\mathbb{Q}$ is modular;
- so there is a map $\varphi\colon X_0(N) \to E$ where $N = \mathrm{cond}(E)$ and $X_0(N)$ is the modular curve (defined over $\mathbb{Q}$);
- so we might hope that carefully chosen points $\tau \in X_0(N)$ might map to rational points on $E \ldots$
- Explicitly, φ is given by

$$\varphi(\tau) = -\sum_{n=1}^{\infty} \frac{a_n}{n} q^n \in \mathbb{C}/\Lambda \cong E(\mathbb{C})$$

where $(a_n)_{n=1}^{\infty}$ are the coefficients of both the L-series $L(E, s) = \sum_{n=1}^{\infty} a_n n^{-s}$ and of the modular form $f_E(\tau) = \sum_{n=1}^{\infty} a_n q^n$, where $q = \exp(2\pi i \tau)$.

where $(a_n)_{n=1}^{\infty}$ are the coefficients of both the L-series $L(E,s) = \sum_{n=1}^{\infty} a_n n^{-s}$ and of the modular form $f_E(\tau) = \sum_{n=1}^{\infty} a_n q^n$, where $q = \exp(2\pi i\tau)$.

- We can compute this accurately given enough coefficients a_n provided that $y = \mathrm{Im}(\tau) >> 0$.

where $(a_n)_{n=1}^{\infty}$ are the coefficients of both the L-series $L(E,s) = \sum_{n=1}^{\infty} a_n n^{-s}$ and of the modular form $f_E(\tau) = \sum_{n=1}^{\infty} a_n q^n$, where $q = \exp(2\pi i \tau)$.

- We can compute this accurately given enough coefficients a_n provided that $y = \mathrm{Im}(\tau) >> 0$.

- **but** why should $\varphi(\tau)$ be a **rational** point?

A little more theory

- Points on $X_0(N)$ parametrize triples (E_1, E_2, α) where the E_j are elliptic curves (defined over $\mathbb{C}$) and $\alpha\colon E_1 \to E_2$ is an isogeny with kernel cyclic of order N;

A little more theory

- Points on $X_0(N)$ parametrize triples (E_1, E_2, α) where the E_j are elliptic curves (defined over $\mathbb{C}$) and $\alpha\colon E_1 \to E_2$ is an isogeny with kernel cyclic of order N;

- One recipe to construct such triples is to let K be an imaginary quadratic field, and $\mathfrak{n}$ an ideal such that $\mathbb{Z}_K/\mathfrak{n} \cong \mathbb{Z}/N\mathbb{Z}$ and $E_1 = \mathbb{C}/\mathfrak{a}\mathfrak{n}$, $E_2 = \mathbb{C}/\mathfrak{a}$, with α the natural map.

A little more theory

- Points on $X_0(N)$ parametrize triples (E_1, E_2, α) where the E_j are elliptic curves (defined over $\mathbb{C}$) and $\alpha\colon E_1 \to E_2$ is an isogeny with kernel cyclic of order N;
- One recipe to construct such triples is to let K be an imaginary quadratic field, and $\mathfrak{n}$ an ideal such that $\mathbb{Z}_K/\mathfrak{n} \cong \mathbb{Z}/N\mathbb{Z}$ and $E_1 = \mathbb{C}/\mathfrak{a}\mathfrak{n}$, $E_2 = \mathbb{C}/\mathfrak{a}$, with α the natural map.
- A sufficient condition for this to be possible is for all prime divisors of N split in K, which we will assume.

A little more theory

- Points on $X_0(N)$ parametrize triples (E_1, E_2, α) where the E_j are elliptic curves (defined over $\mathbb{C}$) and $\alpha\colon E_1 \to E_2$ is an isogeny with kernel cyclic of order N;

- One recipe to construct such triples is to let K be an imaginary quadratic field, and $\mathfrak{n}$ an ideal such that $\mathbb{Z}_K/\mathfrak{n} \cong \mathbb{Z}/N\mathbb{Z}$ and $E_1 = \mathbb{C}/\mathfrak{a}\mathfrak{n}$, $E_2 = \mathbb{C}/\mathfrak{a}$, with α the natural map.

- A sufficient condition for this to be possible is for all prime divisors of N split in K, which we will assume.

- The triple $(\mathbb{C}/\mathfrak{a}\mathfrak{n}, \mathbb{C}/\mathfrak{a}, \alpha) \in X_0(N)(H)$ where H is the Hilbert class field of K. The Galois action is given explicitly by class field theory (giving a very explicit c.f.t. over

imaginary quadratic fields). We get a complete set of Galois conjugates by letting $\mathfrak{a}$ run through representatives of the ideal classes.

imaginary quadratic fields). We get a complete set of Galois conjugates by letting $\mathfrak{a}$ run through representatives of the ideal classes.

- Hence the image of these points under the modular parametrization φ are a complete set of Galois conjugate points in $E(H)$; adding (using the group law) gives a point in $E(K)$.

imaginary quadratic fields). We get a complete set of Galois conjugates by letting $\mathfrak{a}$ run through representatives of the ideal classes.

- Hence the image of these points under the modular parametrization φ are a complete set of Galois conjugate points in $E(H)$; adding (using the group law) gives a point in $E(K)$.

- Provided that E has rank 1 over $\mathbb{Q}$ and also over K, we can take a further trace down to $\mathbb{Q}$ to get a rational point in $E(\mathbb{Q})$.

imaginary quadratic fields). We get a complete set of Galois conjugates by letting $\mathfrak{a}$ run through representatives of the ideal classes.

- Hence the image of these points under the modular parametrization φ are a complete set of Galois conjugate points in $E(H)$; adding (using the group law) gives a point in $E(K)$.

- Provided that E has rank 1 over $\mathbb{Q}$ and also over K, we can take a further trace down to $\mathbb{Q}$ to get a rational point in $E(\mathbb{Q})$.

- the height of this point is given by a formula of Gross and Zagier.

A simpler recipe

To turn the above into a recipe which me may implement simply, we use binary quadratic forms to represent the ideal classes.

$[a, b, c]$ will denote the b.q.f. $aX^2 + bXY + cY^2$; we require $b^2 - 4ac = -D < 0$ and $N \mid a$, so b is a (fixed) root of $b^2 \equiv -D \pmod{4N}$. We then hope to have h forms of the form $[a_i N, b, c_i]$ representing all the h ideal classes and take $\tau_i = (-b + \sqrt{-D})/2a_i N$.

A simpler recipe

To turn the above into a recipe which me may implement simply, we use binary quadratic forms to represent the ideal classes.

$[a, b, c]$ will denote the b.q.f. $aX^2 + bXY + cY^2$; we require $b^2 - 4ac = -D < 0$ and $N \mid a$, so b is a (fixed) root of $b^2 \equiv -D \pmod{4N}$. We then hope to have h forms of the form $[a_i N, b, c_i]$ representing all the h ideal classes and take $\tau_i = (-b + \sqrt{-D})/2a_i N$.

1. Choose a [fundamental] discriminant $-D < 0$ such that $\left(\frac{-D}{p}\right) = +1$ for all $p \mid N$;

2. Choose a root b of $b^2 \equiv -D \pmod{4N}$;

3. Set $c = (b^2 + D)/4N$; check whether the forms $Q_i = [a_i N, b, c_i]$ cover the classes as c_i runs over the divisors of c and $a_i = c/c_i$. If not, choose another b.

4. Use the values $\tau_i = (-b + \sqrt{-D})/2a_i N$.

A few practical remarks

- $\mathrm{Im}(\tau_i) = \sqrt{D}/(2a_iN)$, so we try to maximize the minimum $\sqrt{D}/a_i$. (We cannot change N!)

A few practical remarks

- $\mathrm{Im}(\tau_i) = \sqrt{D}/(2a_iN)$, so we try to maximize the minimum $\sqrt{D}/a_i$. (We cannot change N!)

- We can halve the work by using complex conjugation. The points associated with Q_i, Q_j are conjugate iff $[Q_i * Q_j] = [Q_0]$ where $Q_0 = [N, b, c]$; in each pair we use the one with least a.

A few practical remarks

- $\mathrm{Im}(\tau_i) = \sqrt{D}/(2a_iN)$, so we try to maximize the minimum $\sqrt{D}/a_i$. (We cannot change N!)

- We can halve the work by using complex conjugation. The points associated with Q_i, Q_j are conjugate iff $[Q_i * Q_j] = [Q_0]$ where $Q_0 = [N, b, c]$; in each pair we use the one with least a.

- Delaunay's implementation (but not mine) uses more Galois relations coming from the action of Atkin-Lehner involutions on $X_0(N)$; for square-free N this should reduce the number of τ_i to just **one**.

A few practical remarks

- $\mathrm{Im}(\tau_i) = \sqrt{D}/(2a_iN)$, so we try to maximize the minimum $\sqrt{D}/a_i$. (We cannot change N!)

- We can halve the work by using complex conjugation. The points associated with Q_i, Q_j are conjugate iff $[Q_i * Q_j] = [Q_0]$ where $Q_0 = [N, b, c]$; in each pair we use the one with least a.

- Delaunay's implementation (but not mine) uses more Galois relations coming from the action of Atkin-Lehner involutions on $X_0(N)$; for square-free N this should reduce the number of τ_i to just **one**.

- There is no particular advantage in having the class number h small.

A few practical remarks

- $\mathrm{Im}(\tau_i) = \sqrt{D}/(2a_iN)$, so we try to maximize the minimum $\sqrt{D}/a_i$. (We cannot change N!)

- We can halve the work by using complex conjugation. The points associated with Q_i, Q_j are conjugate iff $[Q_i * Q_j] = [Q_0]$ where $Q_0 = [N, b, c]$; in each pair we use the one with least a.

- Delaunay's implementation (but not mine) uses more Galois relations coming from the action of Atkin-Lehner involutions on $X_0(N)$; for square-free N this should reduce the number of τ_i to just **one**.

- There is no particular advantage in having the class number h small.

- Our implementation uses a recursive scheme to evaluate the sums $\sum_{n=1}^{n_0} a_n q^n$ without having to store a lot of the coefficients a_n; original idea due to Buhler-Gross-Zagier, but with improvements by Cremona-Womack!

Recognising the points

- Each $\varphi(\tau_i) \in \mathbb{C}/\Lambda$, so adding up the conjugates is easy! Then we apply the Weierstrass parametrization map $\mathbb{C}/\Lambda \to E(\mathbb{C})$ (using GP's `ellztopoint()`) to obtain (x, y) coordinates on E, as floating point approximations, which are by construction real and also, in theory, rational.

Recognising the points

- Each $\varphi(\tau_i) \in \mathbb{C}/\Lambda$, so adding up the conjugates is easy! Then we apply the Weierstrass parametrization map $\mathbb{C}/\Lambda \to E(\mathbb{C})$ (using GP's `ellztopoint()`) to obtain (x, y) coordinates on E, as floating point approximations, which are by construction real and also, in theory, rational.

- In simple cases we can use continued fractions (GP's `bestapprox()`) to recover $x \in \mathbb{Q}$ and hence y, as in the first example, so we find $(x, y) \in E(\mathbb{Q})$.

Recognising the points

- Each $\varphi(\tau_i) \in \mathbb{C}/\Lambda$, so adding up the conjugates is easy! Then we apply the Weierstrass parametrization map $\mathbb{C}/\Lambda \to E(\mathbb{C})$ (using GP's `ellztopoint()`) to obtain (x, y) coordinates on E, as floating point approximations, which are by construction real and also, in theory, rational.

- In simple cases we can use continued fractions (GP's `bestapprox()`) to recover $x \in \mathbb{Q}$ and hence y, as in the first example, so we find $(x, y) \in E(\mathbb{Q})$.

- We glossed over one important point: the rational point P constructed is not in general a generator of $E(\mathbb{Q})$ but a multiple of the generator: $P = kP_0$. Luckily, the Gross-Zagier formula gives us an analytic formula (assuming BSD) for this index k. So we divide P by k (on $\mathbb{C}/\Lambda$, before applying `ellztopoint()`) giving k or $2k$ real possibilities to check. (Torsion needs to be handled carefully too here.)

The Cremona-Silverman trick

- We are seeking to recognise a rational point $P_0 = (x_0, y_0) \in E(\mathbb{Q})$ from a floating point approximation to its x-coordinate x_0. We also know the canonical height $\hat{h}(P_0)$ from the BSD formula.

The Cremona-Silverman trick

- We are seeking to recognise a rational point $P_0 = (x_0, y_0) \in E(\mathbb{Q})$ from a floating point approximation to its x-coordinate x_0. We also know the canonical height $\hat{h}(P_0)$ from the BSD formula.

- The canonical height is a sum of local heights, namely

$$\hat{h}(P_0) = h_\infty(x_0) + \log \operatorname{denom} x_0 + \lambda(P_0)$$

where $\lambda(P_0)$ is the sum of contributions from primes where P_0 has bad reduction.

The Cremona-Silverman trick

- We are seeking to recognise a rational point $P_0 = (x_0, y_0) \in E(\mathbb{Q})$ from a floating point approximation to its x-coordinate x_0. We also know the canonical height $\hat{h}(P_0)$ from the BSD formula.

- The canonical height is a sum of local heights, namely

$$\hat{h}(P_0) = h_\infty(x_0) + \log \operatorname{denom} x_0 + \lambda(P_0)$$

  where $\lambda(P_0)$ is the sum of contributions from primes where P_0 has bad reduction.

- Explicit formulas for heights show that the value $\lambda(P_0)$ comes from a **finite set** easily computed from E.

The Cremona-Silverman trick

- We are seeking to recognise a rational point $P_0 = (x_0, y_0) \in E(\mathbb{Q})$ from a floating point approximation to its x-coordinate x_0. We also know the canonical height $\hat{h}(P_0)$ from the BSD formula.

- The canonical height is a sum of local heights, namely

$$\hat{h}(P_0) = h_\infty(x_0) + \log \operatorname{denom} x_0 + \lambda(P_0)$$

where $\lambda(P_0)$ is the sum of contributions from primes where P_0 has bad reduction.

- Explicit formulas for heights show that the value $\lambda(P_0)$ comes from a **finite set** easily computed from E.

- Since we can compute $\hat{h}(P_0)$ using BSD and $h_\infty(x_0)$ from our approximate value of x_0, we can (up to a finite number of possibilities) **compute** the denominator $\operatorname{denom} x_0$ (making use of the fact that it is a perfect square to obtain double precision for free!).

- Since we can compute $\hat{h}(P_0)$ using BSD and $h_\infty(x_0)$ from our approximate value of x_0, we can (up to a finite number of possibilities) **compute** the denominator $\operatorname{denom} x_0$ (making use of the fact that it is a perfect square to obtain double precision for free!).

- this works well in practice, though care is needed: for example, even when we have successfully found $d = \operatorname{denom}(x_0)$ it is not necessarily the case that $\operatorname{num}(x_0)$ is the closest integer to $d\tilde{x}_0$ for our approximation $\tilde{x}_0$ to x_0.

Implementations

The following implementations exist, that I know of. Of course, many people have computed many individual examples; here I include (semi-)automatic packages only.

- My own (a set of GP scripts); Tom Womack also contributed some ideas here; good for curves of conductor up to a million at least;
- Christophe Delaunay's GP scripts;
- Mark Watkins's Magma implementation, originating from Womack's translation of our GP into Magma but now vastly improved (part of Magma distribution since version 2.11);
- Peter Green's GP scripts; not optimized for finding large points.