Explicit isogenies: recent progress and implementations

Luca De Feo

IRMAR, Université de Rennes 1

April 4, 2011
C2, CAES CNRS, Saint-Pierre d'Oléron

Elliptic curves

- Curves of genus 1,
- Abelian varieties of dimension 1.

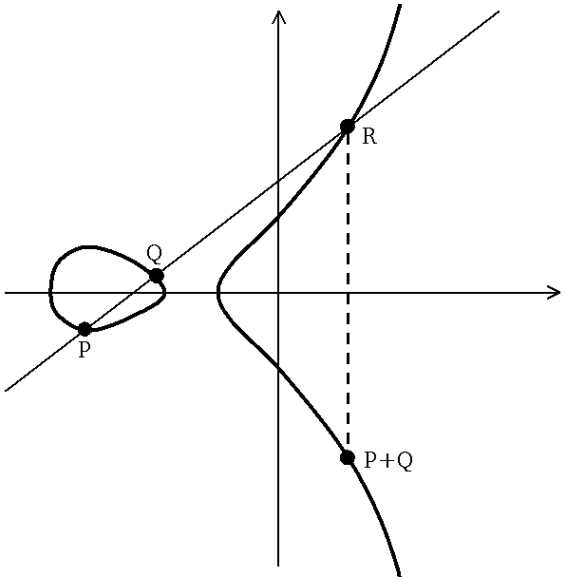
(Short) Weierstrass form

Assuming $p \neq 2, 3$

$$E \ : \ y^2 = x^3 + ax + b \qquad \text{with } a, b \in \mathbb{K}.$$

- discriminant: $\Delta_E = -16(4a^3 + 27b^2) \neq 0$ (the curve is non-singular),
- j-invariant: $j_E = \frac{-1728(4a^3)}{\Delta_E}$ ($j_E = j_{E'} \Leftrightarrow E \cong E'$ over $\bar{K}$),
- invariant differential: $\omega_E = \mathrm{d}\, x/(2y)$ (invariant under translation).

Group law and scalar multiplication



$$y^2 = x^3 + ax + b$$

$$P = (x_0, y_0), Q = (x_1, y_1)$$

$$\lambda = \frac{y_1 - y_0}{x_1 - x_0}$$

$$P + Q = (\lambda^2 - x_0 - x_1, (x_0 - x_2)\lambda - y_0)$$

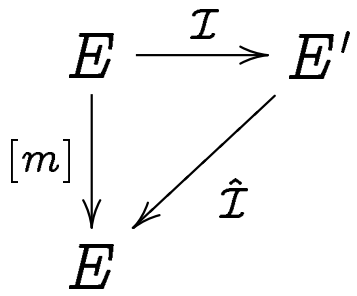
Multiplication: $[m]P = \overbrace{P + P + \cdots + P}^{m \text{ times}}$

m-torsion: $E[m] = \{P \in E(\bar{\mathbb{K}}) | [m]P = \mathcal{O}\} \cong (\mathbb{Z}/m\mathbb{Z})^2$

$$[m](x, y) = \left(\frac{\psi_m(x, y)}{\phi_m^2(x, y)}, \frac{\omega_m(x, y)}{\phi_m^3(x, y)} \right)$$

Division polynomials: ϕ_m can be computed with $O(\log m)$ polynomial multiplications, $\deg \phi_m = O(m^2)$.

Isogenies



(Separable) isogeny: (separable) non-constant rational morphism preserving the identity.

Properties

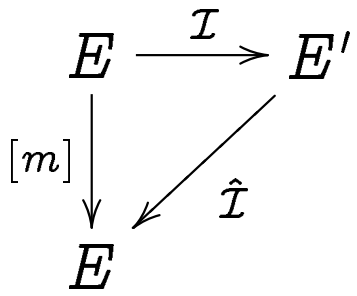
- Isogeny = rational map + group morphism;
- Finite kernel, surjective (in $\bar{\mathbb{K}}$);
- Dual isogeny theorem: they factor the multiplication map into two pieces.

Multiplication

$$[m] : E(\bar{\mathbb{K}}) \to E(\bar{\mathbb{K}})$$
$$P \mapsto [m]P$$

$\ker \mathcal{I} = E[m]$.

Isogenies



(Separable) isogeny: (separable) non-constant rational morphism preserving the identity.

Properties

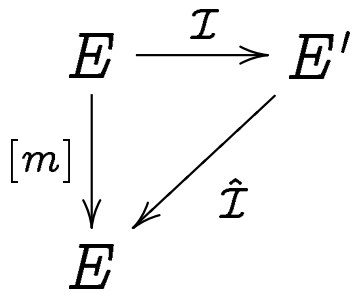
- Isogeny = rational map + group morphism;
- Finite kernel, surjective (in $\bar{\mathbb{K}}$);
- Dual isogeny theorem: they factor the multiplication map into two pieces.

Frobenius endomorphism

$$\varphi : E(\bar{\mathbb{K}}) \to E(\bar{\mathbb{K}})$$
$$(x, y) \mapsto (x^q, y^q)$$

$\ker \varphi = \{\mathcal{O}\}$ (inseparable).

Isogenies



(Separable) isogeny: (separable) non-constant rational morphism preserving the identity.

Properties

- Isogeny = rational map + group morphism;
- Finite kernel, surjective (in $\bar{\mathbb{K}}$);
- Dual isogeny theorem: they factor the multiplication map into two pieces.

Separable isogeny (short Weierstrass form)

$$\mathcal{I}(x,y) = \left(\frac{g(x)}{h(x)}, cy \left(\frac{g(x)}{h(x)} \right)' \right)$$

h vanishes on the abscissas of $\ker \mathcal{I}$. $\deg \mathcal{I} = \# \ker \mathcal{I}$.

Why compute (large) isogenies over finite fields?

SEA algorithm (Schoof 1985; Elkies 1992; Atkin 1988)

Hasse bound $\#E(\mathbb{F}_q) = q - t + 1$;

- **Schoof** Compute t modulo small primes $\ell \Leftrightarrow$ compute the action of φ_q on $E[\ell] \cong (\mathbb{Z}/\ell\mathbb{Z})^2$;
- **Atkin** Determine the order of the roots of $X^2 - tX + q$ by factoring the ℓ-th modular polynomial;
- **Elkies** Compute an ℓ-isogeny $\mathcal{I}$ and the action of φ_q on $\ker \mathcal{I} \cong \mathbb{Z}/\ell\mathbb{Z} \subset E[\ell]$.

Other cryptographic applications

- Transfer DLPs between curves (Gaudry, Hess, and Smart 2002; Smith 2009);
- Construct new cryptosystems (Teske 2006; Rostovtsev and Stolbunov 2006);
- Construct hash functions (Charles, Lauter, and Goren 2009);
- Compute modular polynomials (Bröker, Lauter, and Sutherland 2010);
- Compute the endomorphism ring (Kohel 1996; Bisson and Sutherland 2011).

Vélu's formulas

Compute an isogeny with given kernel (Vélu 1971)

Given the kernel H, computes $\mathcal{I} : E \to E/H$ given by

$$\mathcal{I}(\mathcal{O}_E) = \mathcal{I}(\mathcal{O}_{E/H}),$$

$$\mathcal{I}(P) = \left(x(P) + \sum_{Q \in H^*} x(P+Q) - x(Q), y(P) + \sum_{Q \in H^*} y(P+Q) - y(Q) \right).$$

In practice, given $h(x)$, of degree $\ell - 1$, vanishing on H

$$y^2 = f(x), \qquad p_1 = \sum_{Q \in H^*} x(Q), \qquad \frac{g(x)}{h(x)} = \ell x - p_1 - f'(x)\frac{h'(x)}{h(x)} - 2f(x)\left(\frac{h'(x)}{h(x)}\right)'$$

$$\mathcal{I}(x,y) = \left(\frac{g(x)}{h(x)}, y\left(\frac{g(x)}{h(x)}\right)' \right)$$

Modular polynomial

$\Phi_\ell(X, Y)$, the minimal polynomial over $\mathbb{C}$ of the modular function $j(\ell\tau)$

Properties

- The roots of $\Phi_\ell(X, j(E))$ are the j-invariants of the elliptic curves ℓ-isogenous to E;
- Symmetric in X and Y, degree $\ell + 1$;
- Integer coefficients of size $O(\ell \log \ell)$.

Computation

- By evaluation-interpolation over $\mathbb{C}$ in $\tilde{O}(\ell^3)$ (Enge 2009),
- $\Phi_\ell \bmod p$ in $\tilde{O}(\ell^2 \log p)$ only for special p's,
- By CRT $\Phi_\ell \bmod m$ in $\tilde{O}(\ell^3)$ using only $\tilde{O}(\ell^2 \log m)$ space by CRT (Bröker, Lauter, and Sutherland 2010).

Computing the kernel of an isogeny

Normalized isogenies

An isogeny $\mathcal{I} : E \to E'$ induces an action on the differentials:

$$\mathcal{I}^* \omega_{E'} = c\omega_E \qquad \text{with } c \in \mathbb{K}.$$

Then

$$(cy\mathcal{I}_x(x)')^2 = \mathcal{I}_x(x)^3 + a'\mathcal{I}_x(x) + b'. \tag{1}$$

When $\mathcal{I}^* \omega_{E'} = \omega_E$, the isogeny is said to be normalized.

Algorithm (Elkies 1998; Bostan, Morain, Salvy, and Schost 2008)

1. Factor $\Phi_\ell(X, j_E)$ to obtain an ℓ-isogenous j-invariant $j_{E'}$; $\tilde{O}(\ell^3)$
2. Compute a normalized model for E'; $\tilde{O}(\ell^3)$
3. Solve the differential equation (1). $\tilde{O}(\ell)$

Steps 1 and 2 can be replaced by an algorithm to evaluate large degree isogenies with complexity $O\left(L_q(1/2)\log \ell\right)$ (Jao and Soukharev 2010).

Computing the kernel of an isogeny

Finite fields of small characteristic (Lercier and Sirvent 2008)

1. Factor $\Phi_\ell(X, j_E)$ in $\mathbb{F}_q$ to obtain an ℓ-isogenous j-invariant $j_{E'}$; $\tilde{O}(\ell^3)$
2. Lift j_E and $j_{E'}$ in $\mathbb{Q}_q$ so that $\Phi_\ell(\tilde{j}_E, \tilde{j}_{E'}) = 0$ $\tilde{O}(\ell)$
3. Compute a normalized model for the lift of E'; $\tilde{O}(\ell^3)$
4. Solve the differential equation (1) in $\mathbb{Q}_q$; $\tilde{O}(\ell)$
5. Reduce in $\mathbb{F}_q$. $\tilde{O}(\ell)$

Computing the kernel of an isogeny

Finite fields of small characteristic (Lercier and Sirvent 2008)

1. Factor $\Phi_\ell(X, j_E)$ in $\mathbb{F}_q$ to obtain an ℓ-isogenous j-invariant $j_{E'}$; $\tilde{O}(\ell^3)$
2. Lift j_E and $j_{E'}$ in $\mathbb{Q}_q$ so that $\Phi_\ell(\tilde{j}_E, \tilde{j}_{E'}) = 0$ $\tilde{O}(\ell)$
3. Compute a normalized model for the lift of E'; $\tilde{O}(\ell^3)$
4. Solve the differential equation (1) in $\mathbb{Q}_q$; $\tilde{O}(\ell)$
5. Reduce in $\mathbb{F}_q$. $\tilde{O}(\ell)$

For $p = 2$ step 4 takes $O(\ell^2)$.
This constitutes a bottleneck in practice.

How to solve the differential equation

$$(y\mathcal{I}'_x)^2 = \mathcal{I}_x^3 + a\mathcal{I}_x + b$$

The initial condition in $x = 0$ is unknown, but $\mathcal{I}(\mathcal{O}_E) = \mathcal{O}_{E'}$. Set

$$T = \frac{1}{\mathcal{I}_x(1/x)}, \quad \text{then} \qquad P(T, x) = 0, \qquad T = x + O(x^2)$$

Note: the original paper uses $S = \sqrt{T(x^2)}$. Our choice saves a constant factor.

Quadratic iteration

Let $T = \sum_i t_i x^i$, then

$$(2i-1)t_i x^{i-1} = P(T, x) + O(x^i)$$

- At least one p-adic digit lost every p iterations,
- A total of $O(\ell/p)$ digits is lost.

Newton iteration

Only $\log^2 \ell$ digits are lost in total:

- All intermediate computations lie over $\mathbb{F}_q[[x]]$;
- Only one integral at each iteration;
- At the i-th iteration, divisions by at most $p^{O(i)}$ occur.

What breaks when $p = 2$?

Divisions by 2

Curves have equation $y^2 + xy = x^3 + b$, $\mathcal{I}$ satisfies

$$\left(x^3 - x^2/4 + b\right) \mathcal{I}_x'^2 = \mathcal{I}_x^3 - \mathcal{I}_x^2/4 + b'.$$

This seems avoidable.

Square roots

The Newton iteration is

$$T_{i+1} = T_i + T_i' \sqrt{bx^3 + x/4 + 1} \sqrt{x} \int \frac{P(T_i, x)}{2 T_i'^2 \sqrt{bx^3 + x/4 + 1}^3} \frac{1}{\sqrt{x}}.$$

This seems a more fundamental problem, due the factor $\mathcal{I}_x'^2$ in the equation.

Alternatives?

- Reduce the differential equation modulo 2 and find the coefficients of T by solving a linear system (Lercier 1996). $O(\ell^\omega)$, but fast in practice.
- The algorithms I will present next. $\tilde{O}(\ell^2)$ in the best case.

Algorithms independent from the degree

- Computing Φ_ℓ is the most expensive step. Even if we are given E, E' ℓ-isogenous, we still need Φ_ℓ to compute ℓ-normalized models.
- Suppose we are given E, E' and a bound n on the isogeny degree.

Couveignes' algorithms (Couveignes 1994; Couveignes 1996)

Only for ordinary curves over finite fields:

1. Construct $E[p^k]$ and $E'[p^k]$ for $p^k \ll n$,
2. Pick up generators P and P' of $E[p^k]$ and $E'[p^k]$ respectively,
3. Interpolate the algebraic map

$$\begin{aligned} f \ : \ E[p^k] &\to E'[p^k] \\ [i]P &\mapsto [i]P' \end{aligned}$$

4. Test if f is an isogeny $E \to E'$. If not, choose different P and P'.

The test can be done simultaneously for any $\ell < n$ using a fast XGCD algorithm (Khodadad and Monagan 2006).

Algorithms independent from the degree

Couveignes 1994

- Works in the formal groups of E and E';
- Mainly computations on power series;
- Implemented in (Lercier 1997);
- Complexity $O(\ell^3)$.
- Possibly improvable to $\tilde{O}(\ell^2)$.

Couveignes 1996

- Uses a p-descent in the Weierstrass model by Voloch 1990;
- Computations in towers of Artin-Schreier extensions over $\mathbb{F}_q$;
- Optimized and implemented in (De Feo and Schost 2009; De Feo 2011);
- Quasi-optimal complexity $\tilde{O}(\ell^2)$;
- Practical for $p = 2, 3$.

Downside: both algorithms have an exponential dependency in $\log p$.

Implementations

Done

- Arithmetics for Artin-Schreier towers: `C++`, GPL'ed, available at `http://www.lix.polytechnique.fr/~defeo/FAAST`.
- Couveignes 1996: `C++`;
- Bostan, Morain, Salvy, and Schost 2008; Lercier and Sirvent 2008: MAGMA.

The last two are not available on the net. Ask-me if you are in a rush to use them.

Ongoing implementation in Sage

- Already in: Vélu formulae, Stark 1973, part of Bostan, Morain, Salvy, and Schost 2008. Thanks to Dan Shumow.
- Implementing all the previous algorithms;
- Porting FAAST (will take longer).

Open problems

- Give an algorithm to compute $\Phi_\ell \bmod p^k$ in optimal time.
- Improve (Lercier and Sirvent 2008) in characteristic 2.
- Formally prove an equivalence between (Couveignes 1994) and (Couveignes 1996).
- Find other algorithms independent from the degree, with a polynomial dependency in $\log p$.
- Generalize to genus 2 and higher.