Rational isogenies

Computing rational isogenies from the equations of the kernel

David Lubicz, Damien Robert

1

Theta functions

Complex abelian varieties

- Abelian variety over $\mathbb{C}$: $A = \mathbb{C}^g / (\mathbb{Z}^g + \Omega\mathbb{Z}^g)$, where $\Omega \in \mathscr{H}_g(\mathbb{C})$ the Siegel upper half space.
- The theta functions with characteristic are analytic (quasi periodic) functions on $\mathbb{C}^g$.

$$\vartheta \left[\begin{smallmatrix} a \\ b \end{smallmatrix} \right] (z, \Omega) = \sum_{n \in \mathbb{Z}^g} e^{\pi i \, {}^t(n+a)\Omega(n+a) + 2\pi i \, {}^t(n+a)(z+b)} \qquad a, b \in \mathbb{Q}^g$$

Quasi-periodicity:

$$\vartheta \left[\begin{smallmatrix} a \\ b \end{smallmatrix} \right] (z + m_1\Omega + m_2, \Omega) = e^{2\pi i({}^t a \cdot m_2 - {}^t b \cdot m_1) - \pi i \, {}^t m_1 \Omega m_1 - 2\pi i \, {}^t m_1 \cdot z} \vartheta \left[\begin{smallmatrix} a \\ b \end{smallmatrix} \right] (z, \Omega).$$

- Projective coordinates:

$$\begin{array}{ccc} A & \longrightarrow & \mathbb{P}^{n^g - 1}_{\mathbb{C}} \\ z & \longmapsto & (\vartheta_i(z))_{i \in Z(\overline{n})} \end{array}$$

where $Z(\overline{n}) = \mathbb{Z}^g / n\mathbb{Z}^g$ and $\vartheta_i = \vartheta \left[\begin{smallmatrix} 0 \\ \frac{i}{n} \end{smallmatrix} \right] (., \frac{\Omega}{n})$.

Theta functions of level n

- Translation by a point of n-torsion:

$$\vartheta_i(z + \frac{m_1}{n}\Omega + \frac{m_2}{n}) = e^{-\frac{2\pi i}{n}\ {}^t i\cdot m_1}\vartheta_{i+m_2}(z).$$

- $(\vartheta_i)_{i\in Z(\overline{n})}$: basis of the theta functions of level n
  $\Leftrightarrow A[n] = A_1[n] \oplus A_2[n]$: symplectic decomposition.

- $(\vartheta_i)_{i\in Z(\overline{n})} = \begin{cases} \text{coordinates system} & n \geqslant 3 \\ \text{coordinates on the Kummer variety } A/\pm 1 & n = 2 \end{cases}$

- Theta null point: $\vartheta_i(0)_{i\in Z(\overline{n})} =$ modular invariant.

Riemann Relations

Theorem (Koizumi–Kempf)

Let F be a matrix of rank r such that ${}^tFF = \ell\,\mathrm{Id}_r$. Let $X \in (\mathbb{C}^g)^r$ and $Y = F(X) \in (\mathbb{C}^g)^r$. Let $j \in (\mathbb{Q}^g)^r$ and $i = F(j)$. Then we have

$$\vartheta\begin{bmatrix}0\\ i_1\end{bmatrix}(Y_1, \frac{\Omega}{n})\dots\vartheta\begin{bmatrix}0\\ i_r\end{bmatrix}(Y_r, \frac{\Omega}{n}) = \sum_{\substack{t_1,\dots,t_r \in \frac{1}{\ell}\mathbb{Z}^g/\mathbb{Z}^g \\ F(t_1,\dots,t_r)=(0,\dots,0)}} \vartheta\begin{bmatrix}0\\ j_1\end{bmatrix}(X_1+t_1, \frac{\Omega}{\ell n})\dots\vartheta\begin{bmatrix}0\\ j_r\end{bmatrix}(X_r+t_r, \frac{\Omega}{\ell n}),$$

(This is the isogeny theorem applied to $F_A : A^r \to A^r$.)

- If $\ell = a^2 + b^2$, we take $F = \begin{pmatrix} a & b \\ -b & a\end{pmatrix}$, so $r = 2$.
- In general, $\ell = a^2 + b^2 + c^2 + d^2$, we take F to be the matrix of multiplication by $a + bi + cj + dk$ in the quaternions, so $r = 4$.

The differential addition law ($k = \mathbb{C}$)

$$\Big(\sum_{t\in Z(\bar{2})} \chi(t)\vartheta_{i+t}(x+y)\vartheta_{j+t}(x-y)\Big).\Big(\sum_{t\in Z(\bar{2})} \chi(t)\vartheta_{k+t}(0)\vartheta_{l+t}(0)\Big) =$$

$$\Big(\sum_{t\in Z(\bar{2})} \chi(t)\vartheta_{-i'+t}(y)\vartheta_{j'+t}(y)\Big).\Big(\sum_{t\in Z(\bar{2})} \chi(t)\vartheta_{k'+t}(x)\vartheta_{l'+t}(x)\Big).$$

where $\chi \in \hat{Z}(\bar{2}), i, j, k, l \in Z(\bar{n})$

$$(i', j', k', l') = F(i, j, k, l)$$

$$F = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Example: addition in genus 1 and in level 2

Differential Addition Algorithm:
Input: $P = (x_1 : z_1)$, $Q = (x_2 : z_2)$
and $R = P - Q = (x_3 : z_3)$ with $x_3 z_3 \neq 0$.
Output: $P + Q = (x' : z')$.

1. $x_0 = (x_1^2 + z_1^2)(x_2^2 + z_2^2)$;
2. $z_0 = \frac{A^2}{B^2}(x_1^2 - z_1^2)(x_2^2 - z_2^2)$;
3. $x' = (x_0 + z_0)/x_3$;
4. $z' = (x_0 - z_0)/z_3$;
5. Return $(x' : z')$.

2

Computing isogenies (geometrically)

The isogeny formula

$$\ell \wedge n = 1, \quad A = \mathbb{C}^g/(\mathbb{Z}^g + \Omega\mathbb{Z}^g), \quad B = \mathbb{C}^g/(\mathbb{Z}^g + \ell\Omega\mathbb{Z}^n)$$

$$\vartheta_b^A := \vartheta \begin{bmatrix} 0 \\ \frac{b}{n} \end{bmatrix} \left(\cdot, \frac{\Omega}{n}\right), \quad \vartheta_b^B := \vartheta \begin{bmatrix} 0 \\ \frac{b}{n} \end{bmatrix} \left(\cdot, \frac{\ell\Omega}{n}\right)$$

Proposition

Let F be a matrix of rank r such that ${}^tFF = \ell\,\mathrm{Id}_r$. Let X in $(\mathbb{C}^g)^r$ and $Y = XF^{-1} \in (\mathbb{C}^g)^r$. Let $i \in (Z(\overline{n}))^r$ and $j = iF^{-1}$. Then we have

$$\vartheta_{i_1}^B(Y_1)\dots\vartheta_{i_r}^B(Y_r) = \sum_{\substack{t_1,\dots,t_r \in \frac{1}{\ell}\mathbb{Z}^g/\mathbb{Z}^g \\ (t_1,\dots,t_r)F=(0,\dots,0)}} \vartheta_{j_1}^A(X_1+t_1)\dots\vartheta_{j_r}^A(X_r+t_r),$$

Corollary

$$\vartheta_k^B(0)\vartheta_0^B(0)\dots\vartheta_0^B(0) = \sum_{\substack{t_1,\dots,t_r \in K \\ (t_1,\dots,t_r)F=(0,\dots,0)}} \vartheta_{j_1}^A(t_1)\dots\vartheta_{j_r}^A(t_r), \quad (j = (k,0,\dots,0)F^{-1} \in Z(\overline{n}))$$

Normalizing points

- The isogeny formula assume that the points are in affine coordinates. In practice, given $A/\mathbb{F}_q$ we only have projective coordinates $\Rightarrow$ we need to normalize the coordinates;
- Let P be a projective point on A, and $\widetilde{P}$ be any lift. Note $\widetilde{P} = \lambda \widetilde{P}_0$ where $\widetilde{P}_0$ is a good lift (coming from the affine theta functions);
- If $\ell = 2m+1$, we have that $\vartheta_i((m+1)\widetilde{P}_0) = \vartheta_{-i}(m\widetilde{P}_0)$;
- By computing $m\widetilde{P}, (m+1)\widetilde{P}$ using differential additions, we recover an equation $\lambda^\ell = \mu$. We call P a potential good lift if $\mu = 1$.

Theorem ([LR12])

Let $e_1, \dots e_g$ be a basis of the maximal isotropic kernel K. Assume that we have chosen a potential good lift for $\widetilde{e_i}, \widetilde{e_i + e_j}$. Then

- *Up to an action of $\mathrm{Sp}_{2g}(\mathbb{Z})$, the $\widetilde{e_i}, \widetilde{e_i + e_j}$ are good lifts;*
- *If all points in K are then computed using the Riemann relations, they are also good lifts.*

Working in the algebra $\{\lambda_i^\ell = \mu_i\}$

- Taking potential good lift involve computing ℓ-roots;
- But by [CR11], each choice give the same final results;
- More precisely, the isogeny formulas only involves the λ_i^ℓ;

$\Rightarrow$ We can compute the isogeny over the field defined by the geometric points of the kernel.

The algorithm

H hyperelliptic curve of genus 2 over $k = \mathbb{F}_q$, $J = \mathrm{Jac}(H)$, ℓ odd prime, $2\ell \wedge \mathrm{char}\, k = 1$. Compute all rational (ℓ, ℓ)-isogenies $J \mapsto \mathrm{Jac}(H')$ (we suppose the zeta function known):

1. Compute the extension $\mathbb{F}_{q^n}$ where the geometric points of the maximal isotropic rational kernels of $J[\ell]$ lives.
2. Compute a "symplectic" basis of $J[\ell](\mathbb{F}_{q^n})$.
3. Find all rational maximal isotropic kernels K.
4. For each such kernel K, convert its basis from Mumford to theta coordinates of level 2 (Rosenhain then Thomae).
5. Compute the other points in K in theta coordinates using differential additions.
6. Apply the change level formula to recover the theta null point of J/K.
7. Compute the Igusa invariants of J/K ("Inverse Thomae").
8. Distinguish between the isogeneous curve and its twist.

Complexity over $\mathbb{F}_q$

- The geometric points of the kernel live in a extension k' of degree at most $\ell^g - 1$ over $k = \mathbb{F}_q$;
- Computing the normalization factor takes $O(\log \ell)$ operations in k';
- Computing the points of the kernel via differential additions take $O(\ell^g)$ operations in k';
- If $\ell \equiv 1 \pmod 4$, applying the isogeny formula take $O(\ell^g)$ operations in k';
- If $\ell \equiv 3 \pmod 4$, applying the isogeny formula take $O(\ell^{2g})$ operations in k';

$\Rightarrow$ The total cost is $\widetilde{O}(\ell^{2g})$ or $\widetilde{O}(\ell^{3g})$ operations in $\mathbb{F}_q$.

Remark

The complexity is much worse over a number field because we need to work with extensions of much higher degree.

3

Computing isogenies (rationally)

Equations of the Kernel

- We suppose that we have (projective) equations of K in diagonal form over the base field k:

$$P_1(X_0, X_1) = 0$$

$$\dots$$

$$X_n X_0^d = P_n(X_0, X_1)$$

- By setting $X_0 = 1$ we can work with affine coordinates. The projective solutions can be written $(x_0, x_0x_1, \dots, x_0x_n)$ so X_0 can be seen as the normalization factor.
- Note: I don't know how to obtain equations of K without computing the geometric points of K as we don't have modular polynomials in higher dimension (yet).

Operations on generic points

- We can work in the algebra $\mathfrak{A} = k[X_1]/(P_1(X_1))$, each operation takes $\widetilde{O}(\ell^g)$ operations in k (this is also "true" for number fields).
- A generic point is $\eta = (X_0, X_0X_1, X_0P_2(X_1), \ldots, X_0P_n(X_1))$;
- By computing differential additions over the algebra $\mathfrak{A}$, one can recover a generic normalization $X_0^\ell = \mu \in \mathfrak{A}$;
- We assume here that none of the coordinates of the geometric points are zero, otherwise computing generic differential additions get tricky;
- If we suppose P_1 irreducible, the Galois action on η give "linearly free irreducible points".

The generic algorithm (first version)

- Use the Galois action to compute g "linearly independent generic points" $\eta_1, \ldots, \eta_g$;
- Compute the $\eta_i + \eta_j$ over $\mathfrak{A}$;
- Normalize each of these points;
- Use differential additions to formally compute each points of the kernel;
- Apply the isogeny formula. The result is computed in $\mathfrak{A}$ but will actually be in k.

Remark

This look nice, but in fact this is just a fancy way of working over the splitting field of P_1. In this case we can as well work directly with the geometric points of K so we gain nothing!

Uniform normalization

- Normalizing the basis and using differential addition to compute the rest of the kernel assure that we have a uniform normalization;
- But in the equation

$$\vartheta_k^B(0)\vartheta_0^B(0)\dots\vartheta_0^B(0) = \sum_{\substack{t_1,\dots,t_r \in K \\ (t_1,\dots,t_r)F=(0,\dots,0)}} \vartheta_{j_1}^A(t_1)\dots\vartheta_{j_r}^A(t_r), \quad (j=(k,0,\dots,0)F^{-1} \in Z(\overline{n}))$$

  we only need to normalize uniformly between the points $t_1,\dots,t_r$ (ie do a local normalization);
- When we work with the geometric points, it's better to normalize only the basis and then use differential addition (which is faster than normal addition) than normalize the points in the kernel independently;
- However here since we do a generic normalization we only need to do it once!

The case $\ell \equiv 1 \pmod 4$

- Let $F = \left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)$. Let $c = -a/b \pmod \ell$. The couple in the kernel of F are of the form (x, cx) for each $x \in K$.
- So we normalize the generic point η, compute $c.\eta$ and then $R := \vartheta^A_{j_1}(\eta)\vartheta^A_{j_2}(c.\eta) \in \mathfrak{A}$.
- We then just have to compute $\sum_{x \in K} R(x_1) \in k$;
- In the euclidean division $XRP_1' = PQ + S$, the result is given by $Q(0)$ (thanks Bill!);
- This last operation is quasi-linear in the degree of $\mathfrak{A}$.

The case $\ell \equiv 3 \pmod 4$

- Essentially the same as before, except the tuple in the kernel of F are of the form $(x_1, x_2, ax_1 + bx_2, cx_1 + dx_2)$ for $(x_1, x_2) \in k^2$;
- we have to work on a plane rather than on a line;
- we need two "independent" generic points, so we work in $\mathfrak{A}^{\otimes 2}$;
- we need three normalizations;
- To evaluate the sum of the final polynomial on the couple of points in the kernel we can apply the preceding formula twice.

Complexity over k

An operation in $\mathfrak{A}$ is $\widetilde{O}(\ell^g)$ operations in k.

- Computing the generic normalization factor takes $\widetilde{O}(\log \ell)$ operations in $\mathfrak{A}$;
- If $\ell \equiv 1 \pmod 4$, working in the line take $O(\log \ell)$ operations in $\mathfrak{A}$;
- If $\ell \equiv 3 \pmod 4$, working in the plane take $O(\log \ell)$ operations in $\mathfrak{A}^{\otimes 2}$;
- The final reduction step is quasi-linear in the degree of the algebra.

$\Rightarrow$ The total cost is $\widetilde{O}(\ell^g)$ or $\widetilde{O}(\ell^{2g})$ operations in k.

Remark

- *If $k = \mathbb{F}_q$ and $\ell \equiv 3 \pmod 4$, it is actually faster to generate equations of the kernel from the geometric point (costing $\widetilde{O}(\ell^{2g})$) and apply the generic algorithm than to use the isogeny formula directly!*
- *Still not quasi-linear in the degree of the isogeny when $\ell \equiv 3 \pmod 4$!*

Perspectives

- Use endomorphisms and not only the multiplication by n to compute the isogeny;
- Need to compute an affine version of the endomorphisms, but the isogeny theorem already gives us that;
- For instance, using the real multipication one can compute cyclic isogenies (this subject will be developed in another PEACE meeting...)

Bibliography

R. Cosset and D. Robert. "An algorithm for computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". Mar. 2011. URL: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. HAL: hal-00578991, eprint: 2011/143 (cit. on p. 11).

D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: *Compositio Mathematica* (July 2012). DOI: 10.1112/S0010437X12000243. arXiv:1001.2016 [math.AG]. URL: http://www.normalesup.org/~robert/pro/publications/articles/isogenies.pdf. HAL: hal-00446062 (cit. on p. 10).