DM 1

Lei Bichang

March 14, 2025

1 Short vertices

Question 1.1. To prove that the bound is tight, we will construct a lattice Λ with $\lambda_1(\Lambda) = \sqrt{\frac{2}{\sqrt{3}}\mathrm{vol}(\Lambda)}$.

Consider the lattice

$$\Lambda := \mathbb{Z}v_1 + \mathbb{Z}v_2, \quad v_1 = (1,0),\ v_2 = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right).$$

For each $x \in \mathbb{Z}$, we have

$$\|v_2 + xv_1\| = \sqrt{\left(x + \frac{1}{2}\right)^2 + \frac{3}{4}} \geq 1 = \|v_2\| = \|v_1\|.$$

So the basis (v_1, v_2) for Λ is LG-reduced. Therefore $\lambda_1(\Lambda) = \|v_1\| = 1$ and $\mathrm{vol}(\Lambda) = \frac{\sqrt{3}}{2}$ as desired.

2 Proving primality

Question 2.1. We shall prove by contadiction. Suppose that N is not a prime.

Let $y^2z = x^3 + ax^2z + bz^3$ be an equation that defines E, $a, b \in \mathbb{Z}/N\mathbb{Z}$, $\Delta(E) = -16(4a^3 + 27b^2) \in (\mathbb{Z}/N\mathbb{Z})^\times$. Let $p \leq \sqrt{N}$ be a prime dividing N, and E_p be the curve over $\mathbb{F}_p$ defined by $y^2z = x^3 + (a \bmod p)x^2z + (b \bmod p)z^3$. Then $\Delta(E_p) = (\Delta(E) \bmod p) \in \mathbb{F}_p^\times$, so E' is an elliptic curve over $\mathbb{F}_p$. Let

$$E(\mathbb{Z}/N\mathbb{Z}) \to E_p(\mathbb{F}_p), \quad Q = [x : y : z] \mapsto \tilde{Q} := [\tilde{x} : \tilde{y} : \tilde{z}]$$

be the reduction modulo p map. It sends O to O, and points in $E(\mathbb{Z}/N\mathbb{Z}) \smallsetminus E^{\mathrm{s}}(\mathbb{Z}/N\mathbb{Z})$ to $E(\mathbb{F}_p) \smallsetminus \{O\}$. Since $2P \notin E^{\mathrm{s}}(\mathbb{Z}/N\mathbb{Z})$ and $2qP = O$, the reduction $\widetilde{2P} \neq O$ and has order q. For the same reason, $qP \in E_p(\mathbb{F}_p)$ has order 2. Hence $\tilde{P} \in E_p(\mathbb{F}_p)$ has order $2q$. By Hasse theorem,

$$2q \leq \#E_p(\mathbb{F}_p) \leq (p+1) + 2\sqrt{p} = (\sqrt{p}+1)^2.$$

However,

$$2q > (N^{\frac{1}{4}} + 1)^2 \geq (\sqrt{p}+1)^2.$$

This is a contadiction.

Question 2.2. Assume that the prime $q \neq 2$, so that $E(\mathbb{F}_p) \simeq \mathbb{Z}/2q\mathbb{Z}$ and there exist $\mathbb{F}_p$-points of order $2q$. There are $q-1$ generators in $\mathbb{Z}/2q\mathbb{Z}$, so a uniformly random point in $\mathbb{Z}/2q\mathbb{Z}$ has the chance of $\frac{q-1}{2q}$ to have order $2q$. Therefore, we can use the following algorithm.

(1) Generate a random point $P \in E(\mathbb{F}_p)$. *Expected cost:* $2p/\#E(\mathbb{F}_p) = p/q$ *times of a constantly many* $\mathbb{F}_p$*-operations.*

(2) Check if $P = O$, P has order 2 (i.e. $P = (x, 0)$ for some x) or P has order q. *Cost: the time of q-multiplication, which is $O(\log q)$ times of $\mathbb{F}_p$-operations.*

(3) If one of the three conditions in step (2) is satified, we go back to step (1). Otherwise P has order $2q$.

The expected time of iterations is $\frac{2q}{q-1}$. So the total complexity is

$$\frac{2q}{q-1}\frac{p}{q}O(\log q) = \frac{2p}{q-1}O(\log q)$$

times of $\mathbb{F}_p$-operations. Since $2q \in [(\sqrt{p}-1)^2, (\sqrt{p}+1)^2]$, the complexity is about

$$\frac{2p}{\frac{p}{2} - \sqrt{p} - \frac{1}{2}}O(\log(\sqrt{p}+1)) \approx O(\log p).$$

Question 2.3. In the other file, I defined a function $\mathsf{test\_conjecture}(p, N)$ that generates N curves over $\mathbb{F}_p$ as in the conjecture and then returns the proportion $P(p)$ of elliptic curves of order $2q$ for some prime q. Then I generated some primes $p_0 < p_1 < \cdots < p_n$, such that $\log(p_{i+1}) - \log(p_i) \approx d$, and repeated the following procedure several times.

(1) For each prime p_i, run $\mathsf{test\_conjecture}(p_i, N)$ and get the proportion $P(p_i)$.

(2) Use linear regression to get c_1, c_2 such that

$$\log P(p) \approx \log c_1 - c_2 \log(\log p).$$

These constants should be larger than the actual constants if the conjecture is true.

After several trials, I would guess that $c_1 = 0.2$, $c_2 = 0.7$.

Question 2.4. We can use the following algorithm.

(1) Pick uniformly random $a, b \in \mathbb{F}_p$ and let $E : y^2 = x^3 + ax + b$ over $\mathbb{F}_p$. If E is not an elliptic curve, redo step (1).

(2) Compute $N := \#E(\mathbb{F}_p)$. If N is divided by 2, go back to step (1). *Cost: polynomial in $\log p$ many $\mathbb{F}_p$-operations.*

(3) Run the Miller-Rabin test k-times to check if $q := N/2$ is a prime. If we find q to be composite, go back to step (1). *Cost: at most $k \cdot O(\log p)$ many $\mathbb{F}_p$-operations.*

(4) Use the algorithm decribed in Question 2.2 to get a point $P \in E(\mathbb{F}_p)[2q]$. *Expected cost: $O(\log p)$ many $\mathbb{F}_p$-operations.*

We run Miller-Rabin test k-times in step (3), so the probability of misclassifying a composite number as prime is 4^{-k}, and q has the chance of $1 - 4^{-k}$ to be a prime.

If Conjecture 1 holds, the expected time of iterations is less than $(\log p)^{c_2}/c_1$. So the expected times of $\mathbb{F}_p$-operations is polynomial in $\log p$.

Question 2.5. We can use the following algorithm, which is a variant of the one in Question 2.4.

(1) Pick uniformly random $a, b \in \mathbb{Z} \cap [0, \ldots, N-1]$ and let $\Delta := -16(4a^3 + 27b^2)$.

- If $\gcd(\Delta, N) \notin \{0, 1\}$, return $\mathsf{composite}$.
- If $\gcd(\Delta, N) = 0$, go back to the start of the algorithm step (1).

- Otherwise define the elliptic curve $E : y^2 = x^3 + ax + b$ over $\mathbb{Z}/N\mathbb{Z}$.

(2) Pretend that N is a prime and use the algorithm for elliptic curves over finite fields to compute $M := \#E(\mathbb{Z}/N\mathbb{Z})$.

- If the program throws an exception, return `composite`.
- Otherwise if M is divided by 4, or $M \leq (N^{\frac{1}{4}} + 1)^2$, go back to step (1).

(3) Run the Miller-Rabin test k-times to check if $q := M/2$ is a prime. If we find q to be composite, go back to step (1).

(4) Pretend that N is a prime and use the algorithm decribed in Question 2.2 to get some $P \in E(\mathbb{Z}/N\mathbb{Z})[2q]$.

- If the program throws an exception, return `composite`.

(5) Use the formulae for elliptic curves over fields to compute $2P$ and qP.

- If the program throws an exception, return `composite`.
- Otherwise return `prime`.

Assume first that N is a prime. By the analysis in Question 2.4, the algorithm cost polynomial time in $\log N$. If N is composite, then the algorithm will terminate with probability $> 1-4^{-k}$ in step (1) - (3), so the cost is smaller than it when N is prime. Therefore, the complexity is polynomial in $\log N$ many $\mathbb{Z}/N\mathbb{Z}$-operations.

Question 2.6. Please see the function `primality_test` in other file. I have implemented most of the functions, but I failed to realize Schoof's algorithm (the computation of trace modulo primes).