Sep 1, 2023 10 min read

Elliptic Curve Point Addition

Updated: Jan 27

This article describes how elliptic curve addition works over real numbers.
Cryptography uses elliptic curves over finite fields, but elliptic curves are easier to conceptualize in a real cartesian plane. This article is aimed at programmers and tries to strike a balance between getting too math heavy and too hand-wavy.

Set theoretic definition of elliptic curves

The set of points on an elliptic curve form a group under elliptic curve point addition.

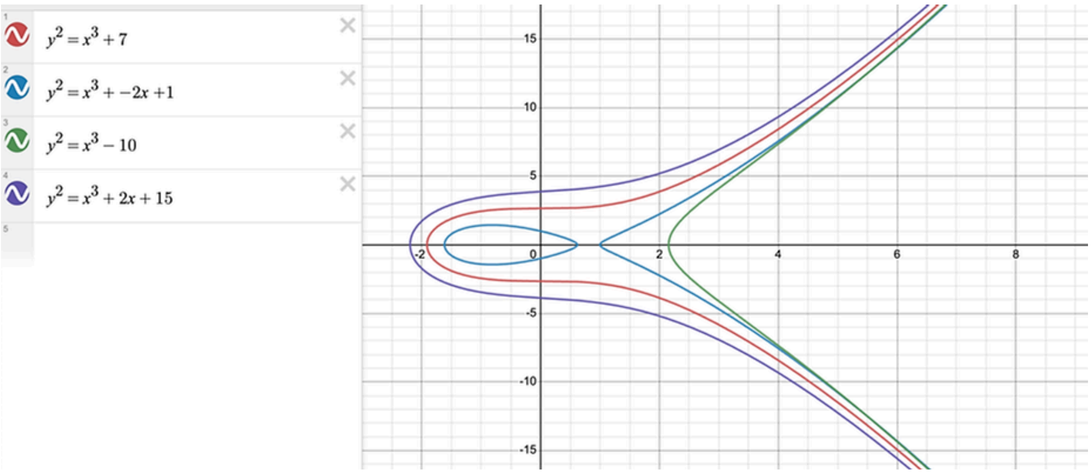
Hopefully, if you’ve been following our group theory introduction, then you actually understood most of this, aside from what “point addition” is. But that’s the beauty of abstract algebra right? You don’t *need* to know what that is, and you still understand the above sentence.

Let’s break it down.

Elliptic curves are a family of curves which have the formula

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

Depending on what value of a and b you pick, you’ll get a curve that looks like some of the following:



A point on an elliptic curve is an (x, y) pair that satisfies $y^2 = x^3 + ax + b$ for a given a and b.

For example, the point (3, 6) is in the curve $y^2 = x^3 + 9$ because it $6^2 = 3^3 + 9$. In group theoretic terms, (3, 6) is a member of the set defined by $y^2 = x^3 + 9$. Since we are dealing with real numbers, the set has infinite cardinality.

The idea here is we can take two points from this set, do a binary operator, and we will get another point that is also in the set. That is, it is an (x, y) pair that also lies on the curve.

Rather than thinking of elliptic curves as a plot on a graph, think of them as an infinite set of points. Points are in the set if and only if they satisfy the elliptic curve equation.

Once we see these points as a set, looking at them as a group isn’t mysterious. We just take two points, and produce a third according to the rules of a group.
Specifically, we need to fulfill

- a binary operator that is closed and associative, i.e. it produces another point in the set

Although we don’t know how the binary operator works, we do know that it takes two (x, y) points on the curve and returns another point on the curve. Because the operator is closed, we know that the point will in fact be a valid solution to the elliptic curve equation, not a point somewhere else.

We also know that this binary operator is associative (and commutative, per the section heading).

So given three points on the elliptic curve A, B, and C, we know the following is true:

- $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- $A \oplus B = B \oplus A$

I’m using $\oplus$, because we know this binary operator is not addition in any normal sense, but a binary operator (again, remember from set theory, a binary operator takes two elements in a set and returns another element in a set, how it does that is not central to the definition).

We also know that there has to be an identity element somewhere. That is, any (x, y) point that falls on the curve is combined with the identity element, the output is the same (x, y) point unchanged.

And because this is a group and not a monoid, every point needs to have an inverse such that $P \oplus P^{-1} = e$, where e is the identity element.

Lastly, because we said it is a group, not a ring or field, we know there isn’t another binary operator to combine points (or at least if there is, we don’t care about it).

The identity element

Intuitively, we might think of (0, 0) or (1, 1) being the identity element, since something like that often is in other groups, but you can see in the plots above that those points generally do not land on the curve. Since they don’t belong to the set of points on $y^2 = x^3 + ax + b$, they are not part of the group.

But recall from set theory that we can define binary operators however we like over sets arbitrarily defined. This allows us to add a special element that isn’t technically on the curve but by definition is the identity element.

I like to think of the identity element as “the point that is nowhere” because if you combine nowhere with any real point, nothing changes. Annoyingly, mathematicians call this point, the identity element “the point at infinity.”

Hey wait, isn’t this point supposed to satisfy $y^2 = x^3 + ax + b$? Nowhere (or infinity) is not a valid value for (x, y).

Ahh, but remember, we can define sets however we like! We define the set that makes up the elliptic curve as points on the elliptic curve *and* the nowhere point.

Because binary operators are just subsets of a cartesian product (a relation), and we can define the relation however we like. We can have as many hacky “if statements” in our arithmetic as we please and still follow the group laws.

Addition is closed.

Without loss of generality, let’s take the elliptic curve

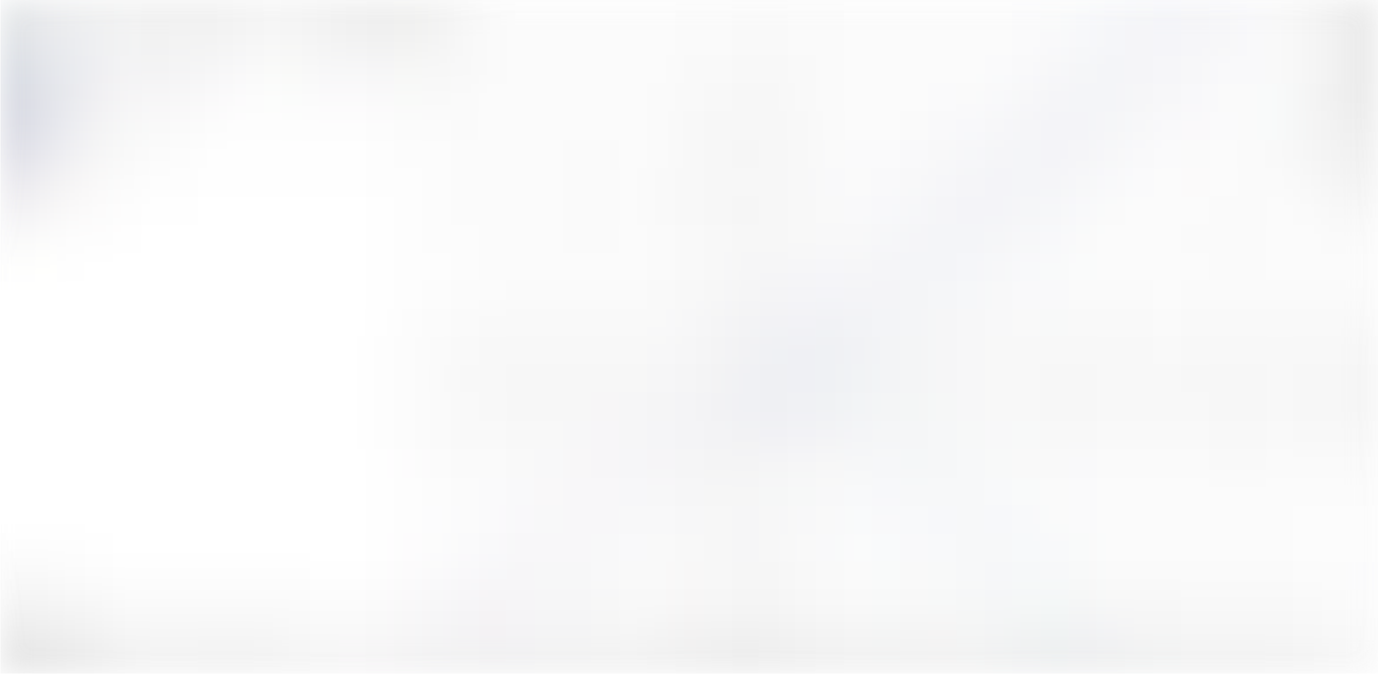
To illustrate how lines intersect on elliptic curves, then let’s draw a nearly vertical line y = 10x

(It could be 1000x to make it more vertical, but we would get numerical instability as you will see later)

We get the following set of plots.

It turns out, even though it looks like the purple line ($y = 10x$) is rising faster than the blue curve ($y^2 = x^3 + 10$), they will always eventually intersect.

If we zoom out far enough, we can see the intersection. This is true in general. As long as x is not "perfectly vertical" it will always intersect with a third point eventually.



To see this mathematically, let's take the square root of both sides of our elliptic curve so we can compare the equations

Even though the second equation gets a big boost from being multiplied by 10, the x in the first equation effectively has a power of 2. In CS terms, we know that $O(n^2)$ will be bigger than O(constant) eventually no matter how big the constant is.

This is true in general. Let's write a general elliptic curve, and a rapidly rising straight line with an arbitrarily large constant behind it

So what we have is an important lemma here: **elliptic curves eventually “rise faster” than straight lines as x gets larger. This means that eventually, the blue curve will “catch up” to the purple line and they will intersect…**

What this says about our above plot is very important:

as long as we do not pick a perfectly vertical line, if we intersect two points in an elliptic curve, then we will also intersect a 3rd point on the elliptic curve.

The “if we intersect two points” is important. If we shift our purple line over to the left so it doesn’t cross the “U-turn” of the elliptic curve, then it will only cross at one point
Another way of understanding it:

If a straight line crosses an elliptic curve at exactly two points, then it must be perfectly vertical.

You could work out an algebraic proof from the formulas above, but I think the geometric argument is more intuitive.

I recommend you stop here and draw some elliptic curves and lines and convince yourself of this visually.

Our exception for vertical lines actually causes inverse and identity elements to fall into place beautifully.

The inverse of an elliptic curve point is the negative of the y value of the pair. That is, the inverse of (x, y) is (x, -y) and vice versa. Drawing a line through such points creates a perfectly vertical line.

The identity element is the “point at infinity” we alluded to earlier is simply the point “way up there” when we draw a vertical line.

Abelian Group

The fact that elliptic curve points are a group under our “2 points always result in a 3rd except for the identity” makes it’s abelian nature obvious.

When we pick two points, there is only one other third point. You can’t get four intersections in an elliptic curve. Since we only have one possible solution, then it is clear that $A \oplus B = B \oplus A$.

Why elliptic curve addition flips over the x axis

We glossed over a very important detail in the last section, because it really deserves a section on its own.

In it’s current form, it has a bug if we add two points where the intersection happens in the middle.

Using our definitions above, the following must be true

- A ⊕ B = C
- A ⊕ C = B
- B ⊕ C = A

With a little algebra, we'll derive a contradiction
(B ⊕ C) ⊕ B = C
B ⊕ C = inv(B) ⊕ C
B = inv(B)

This says B is equal to it's inverse. But B is not the identity element (which is the only element that can be the inverse of itself), so we have a contradiction.

Thankfully, there is a way to rescue this. Just define point addition to be the third point *flipped over the y axis*. Again, we *are allowed to do this* because binary operators can be defined however we like, we just care that our definitions satisfy the group laws.

So the correct way to add elliptic curve points is represented graphically below

Formula for addition

Using some algebra, and given two points

$P_1=(x_1,y_1)$
$P_2=(x_2,y_2)$

One can derive how to compute $P_3 = (x_3, y_3)$ where $P_3 = P_1 \oplus P_2$ using the following formula.

Algebraically demonstrating commutativity and associativity

Because we have a closed form equation, we can prove algebraically that $T \oplus U = U \oplus T$ given points T and U.

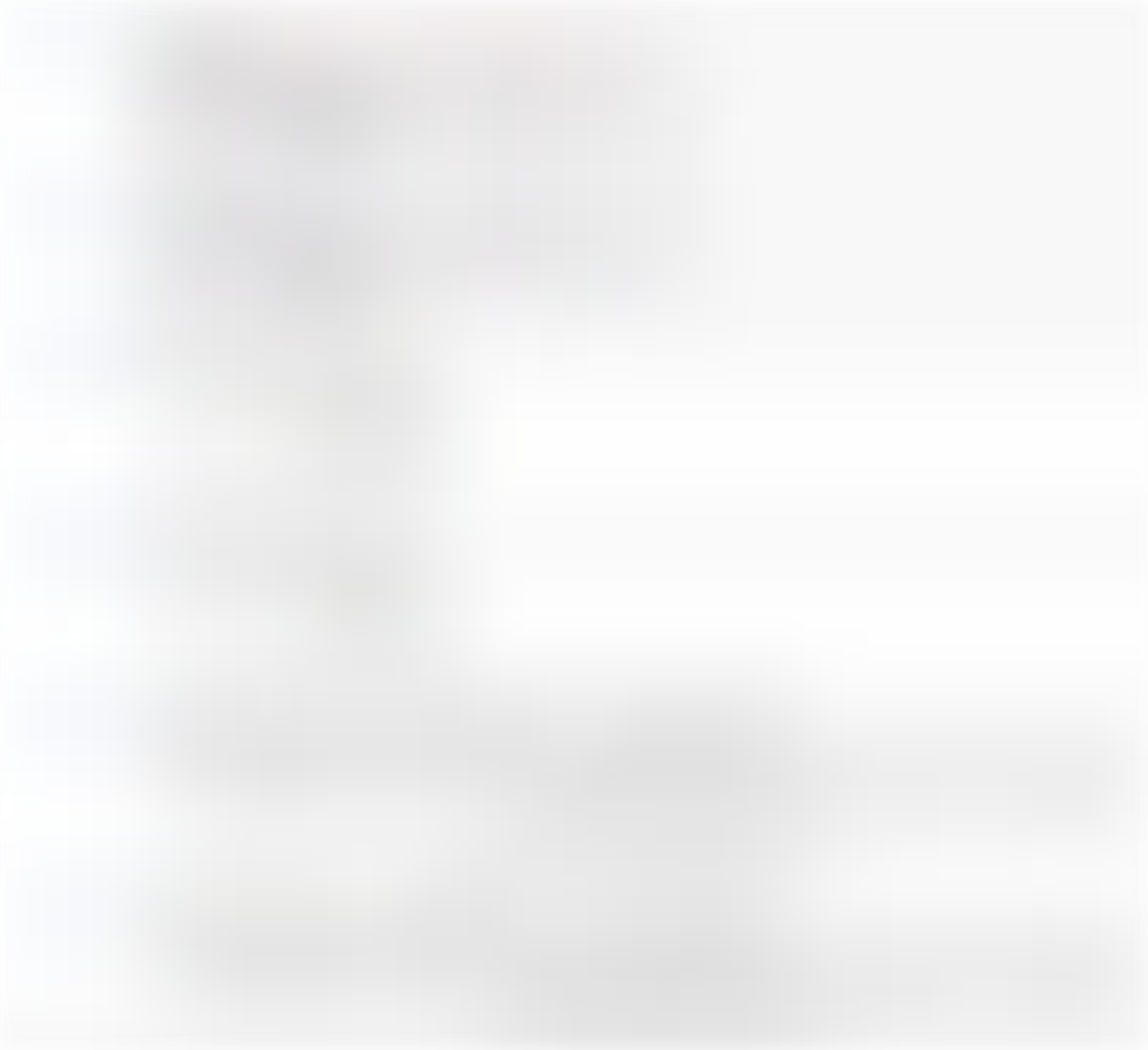
We do it as follows

$P = T \oplus U$
$Q = U \oplus T$

```
var('y_t', 'y_u', 'x_t', 'x_u')
lambda_p = (y_u - y_t)/(x_u - x_t)
x_p = lambda_p^2 - x_t - x_u
y_p = (lambda_p*(x_t - x_p) - y_t)

lambda_q = (y_t - y_u)/(x_t - x_u)
x_q = lambda_q^2 - x_u - x_t
y_q = (lambda_q*(x_u - x_q) - y_u)
```

Here is a screenshot of running the above code in Jupyter notebook and pretty printing the output. The computer algebra system needs a bit of coaxing, but we can clearly see x_q == x_p and y_q == y_p.



P = Q for all (xt, yt) and (xu, yu) values. We get a division by zero error if xt = xu, but this means they are the same point and that is obviously commutative.

We can use similar techniques to demonstrate associativity, but unfortunately this is extremely messy so we refer the interested reader to another proof of associativity.

Elliptic curves meet the abelian group property

2. The group has an identity element.
3. Each point has an inverse.
4. The group is abelian because A ⊕ B = B ⊕ A

A binary operator must accept every possible pair from the set. What if the pair is the same element, i.e. A ⊕ A?

Point multiplication: adding a point with itself

Let’s think of this in limit terms. Adding a point to itself is like bringing two points infintesimally close to each other until they become the same point. When this convergence happens, the slope of the line will lie tangent to the curve.
So adding a point to itself is simply taking the derivative at that point, getting the intersection, then flipping the y axis.

The following image graphically demonstrates A ⊕ A = 2A.

Shortcut for point multiplication

What if we wanted to do 1000A instead of 2A? It would seem this is an $\mathcal{O}$(n) operation, but it isn’t.

Because of associativity, we can write 1000A as

So rather than doing 1000 operations, we can do it in 14 (9 to compute 512, caching the intermediate results, then 5 additions).

This is actually an important property when we get to cryptography:

We can efficiently multiply an elliptic curve point by a large integer efficiently.

Implementation details of addition

It isn't hard to derive the formula for point addition using simple algebra. When we intersect two points, we know the slope and the points that it crossed through, so we can calculate the point of intersection.

I'd rather not do that here because I don't want to get lost in a bunch of symbolic manipulation.

The whole power of group theory is that *we don't care* what that symbolic manipulation looks like. We know that if we do our binary operator on two points, we'll get another point in our set, and our set follows the group laws.

If you think about it that way, elliptic curves are much easier to understand.
Rather than trying to understand elliptic curves in isolation from the ground up, we study a bunch of other algebraic groups, then transfer that knowledge and intuition to elliptic curves.

Rational numbers under addition are a group. Integers modulo a prime are a group under multiplication. Matrices of non-zero determinant under multiplication are a group.

Recent Posts See All

The g

hat tl

latior

t as

ike th

to vis

oint g

Understanding the Function Selector in Solidity

20 0 1

How ERC721 Enumerable Works

178 0 2

happens under the hood when looking at it as a group

Home About Curriculum Pricing Testimonials Tutorials Test Yours APPLY NOW

Tutorials

Instructor Bios

Testimonials

Learn Solidity

Pricing

About RareSkills.

Follow us on

Curriculum

Hire our Developers

Test Yourself

Admission Process and Policy

Contact Us

Privacy Policy