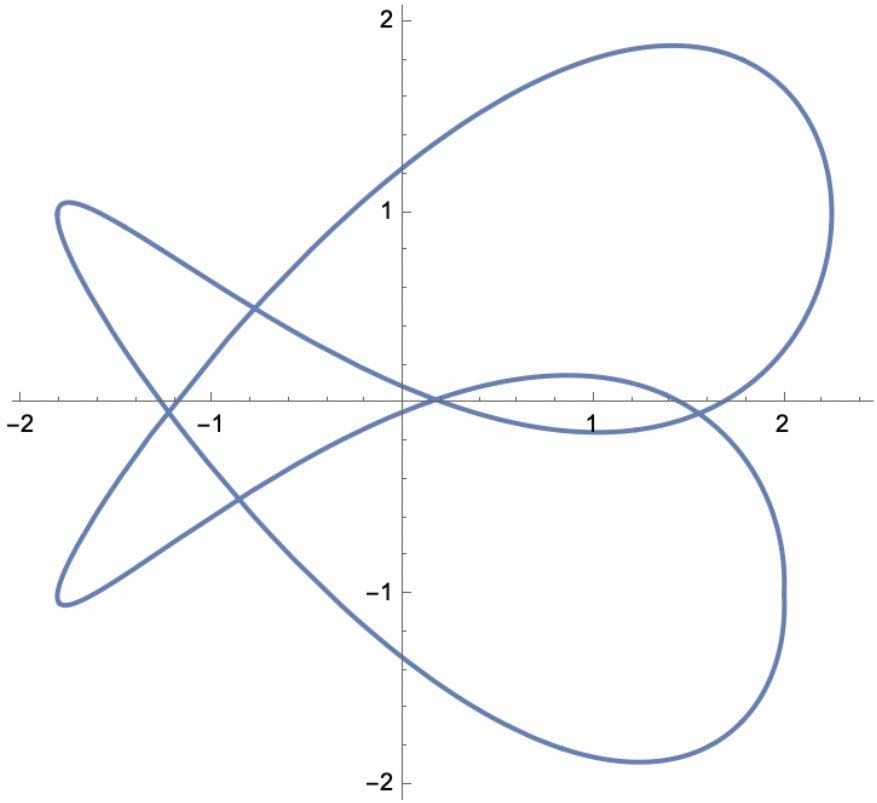
Elliptic curves and ellipses, what's the deal?

Balázs Kőműves

ELTE, Budapest, 22 February 2024

Plane curves

Today[1] we will talk about plane curves[2]. Plane curves are mostly what you would imagine they are: 1 dimensional curves in a 2 dimensional plane. Like for example this one:



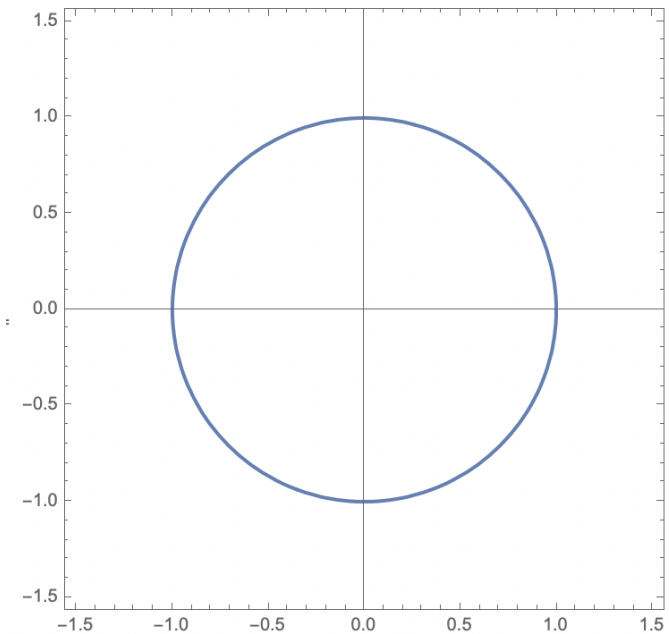
[1] slides available here: `https://github.com/bkomuves/slides/`

[2] and then quite some other stuff...

Algebraic curves

In particular, we will talk about *algebraic curves*, that is, curves defined by an algebraic (or polynomial) equation. Everybody already knows the circle:

$$x^2 + y^2 = 1$$



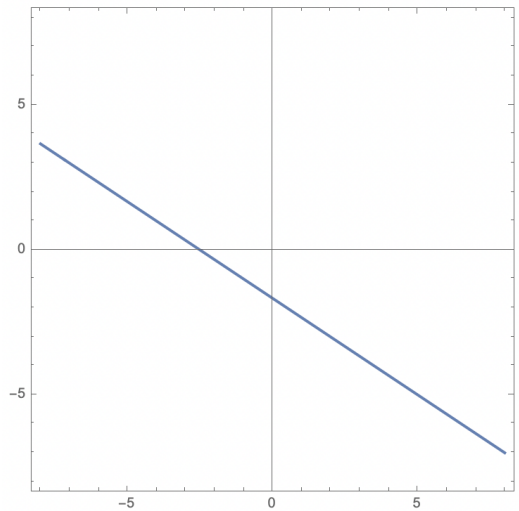
We could write down different equations and study the resulting curves. This is something mathematicians like to do :)

Linear curves (or lines)

The simplest possible equations we can write down are linear (degree 1 polynomials), for example: $2x + 3y + 5 = 0$; or more generally

$$Ax + By + C = 0$$

where A, B, C are some fixed numbers (parameters).



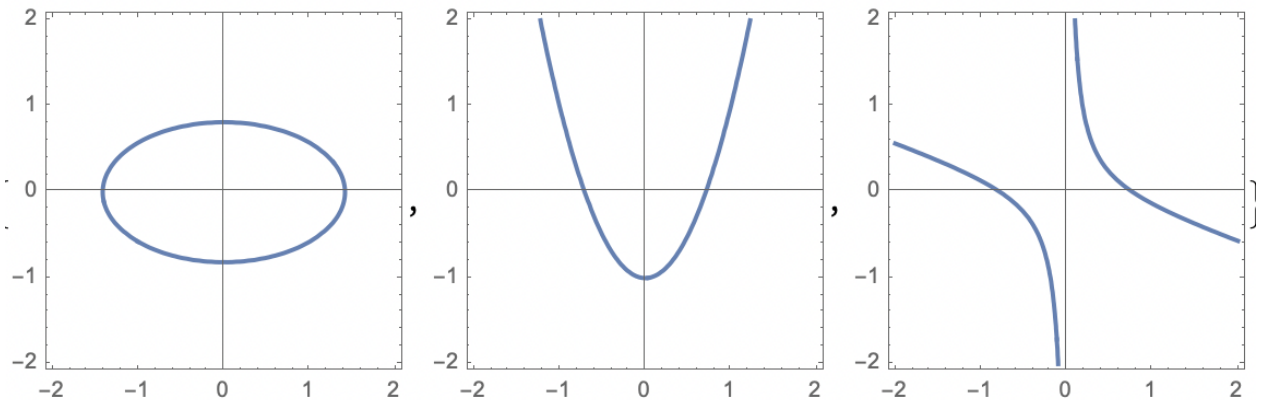
These are just straight lines, we learn them in primary school.

Not much to say about these. An interesting fact that two lines intersect in 1 point, *except* when they are parallel. Don't worry, we will fix that later!

Quadratic curves (or conic sections)

After the linear equation, the next simplest ones are quadratic (degree 2) equations. We learn these in highschool, under the name of "conic sections":

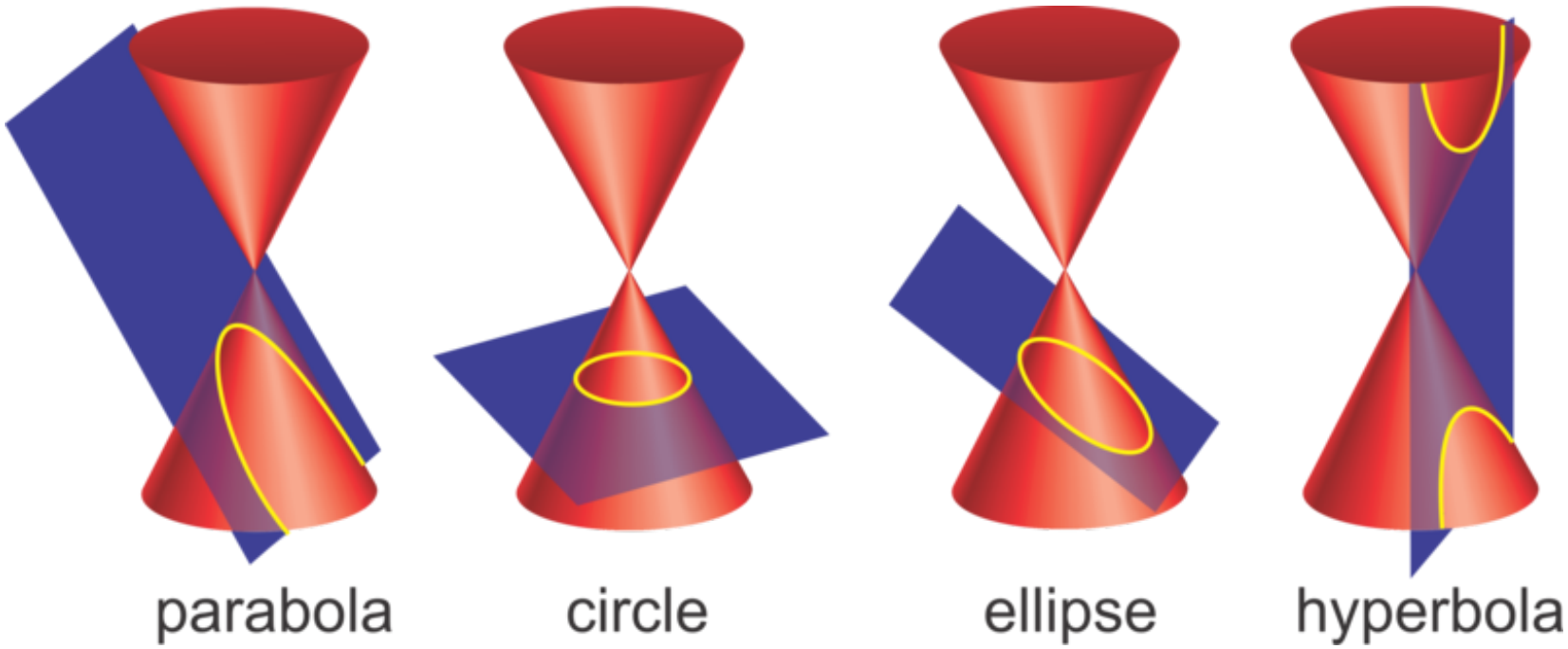
- circle: $x^2 + y^2 = R^2$
- ellipse: $Ax^2 + By^2 = 1$
- parabola: $y = Cx^2$
- hyperbola: $xy = C$



There are two more types though, can you guess?

Conic sections

And this is why they are called "conic sections":

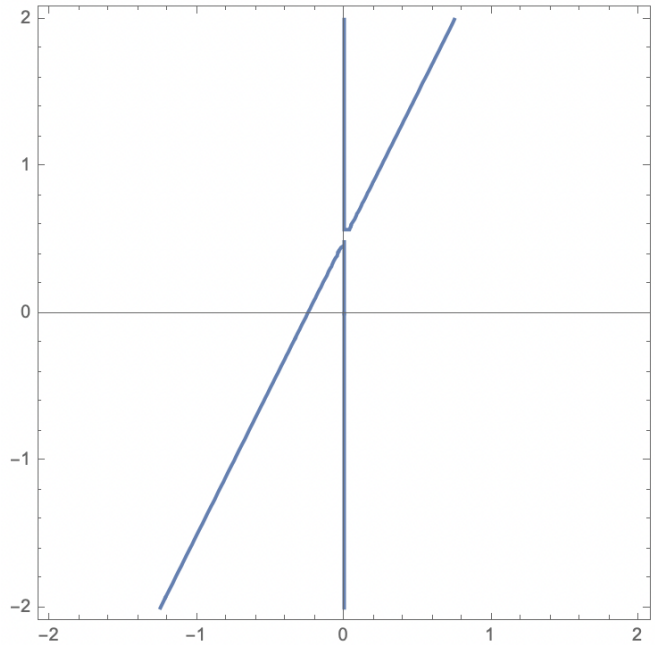


(image source: CK-12)

Singular curves

Two types of quadratic equations are often omitted: $x^2 = 0$ and $xy = 0$. The first one is a "double line", and the second one is two intersecting lines.

```
ContourPlot[x (2 x - y + 1 / 2) == 0, {x, -2, 2}, {y, -2, 2}, Axes → True]
```



These are called "singular", and are less nice than the "non-singular" or "smooth" curves (in the quadratic case also less interesting).

The projective plane $\mathbb{P}^2$

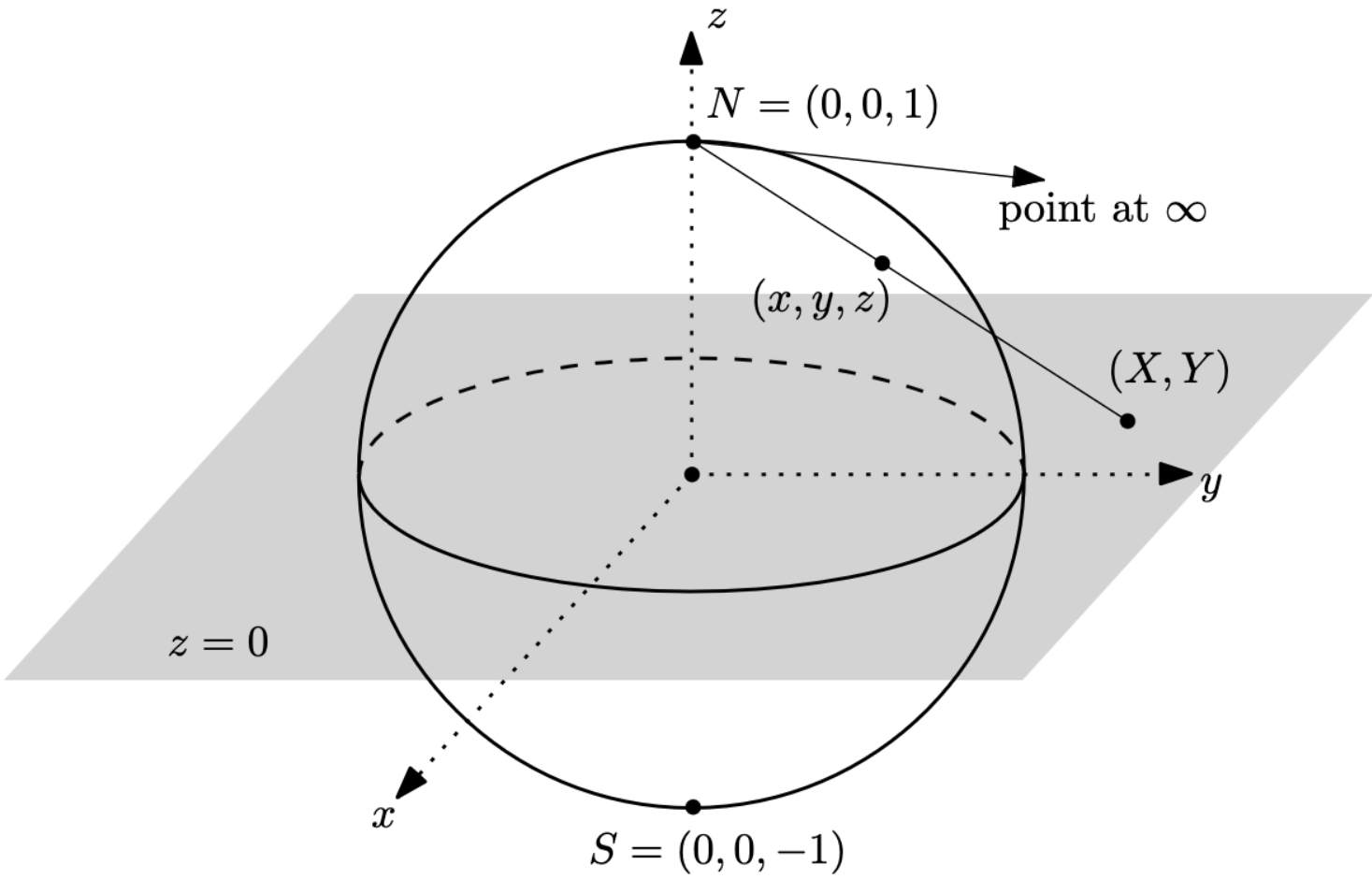
Things become quite a bit nicer if we add some new points "at infinity" to the plane: One infinite point for each line passing through the origin (it's the same infinite in both direction, so the line becomes a circle). This is called the "projective plane", and is a natural "compactification" (meaning roughly that you cannot run away from your problems to infinity anymore :).

You can represent this with homogeneous or projective coordinates: instead of two coordinates (x, y) you will have three coordinates $[X : Y : Z]$; however, now $[X : Y : Z]$ and $[\lambda X : \lambda Y : \lambda Z]$ means the same point (and $[0 : 0 : 0]$ is disallowed). This means that your equations must be homogeneus: For example instead of $2x^3 + 5y^2 = 3$ you have to write $2X^3 + 5Y^2Z = 3Z^3$. You can easily map between the two:

$$(x, y) \mapsto [x : y : 1] \qquad \text{and} \qquad [X : Y : Z] \mapsto (X/Z,\ Y/Z).$$

When $Z = 0$ you get the points at the infinity.

Stereographic projection and the projective plane



The stereographic projection model of the projective plane.

(image source: Adrian Constantin et al)

Conic sections in the projective plane

Let us write down the homegenous equations for the conic sections:

- ellipse: $\alpha X^2 + \beta Y^2 = Z^2$
- parabola: $YZ = \gamma X^2$
- hyperbola: $XY = \gamma Z^2$

But wait! Aren't these just the same?! The last two only differ by permuting the coordinates (a "rotation"), and to get the ellipse, just substitute something like $Y \mapsto Z' - Y'$ and $Z \mapsto Z' + Y'$ (another rotation) into the equation of the parabola.

So in the projective plane, there is no difference between the ellipse, parabola and hyperbola. They only looked different before because they intersect the "line at infinity" differently:

- the ellipse does not intersect it
- the parabola intersects in 1 point (it is tangent to it)
- and the hyperbola intersects in 2 points

But in the projective plane, the "line at infinity" isn't special, it is exactly the same as any other line.

Complex coordinates

Instead of using real numbers as coordinates, we could also use complex numbers (or in fact any other field or even just a ring).

This again makes things a bit nicer, and will become important later. Why does it help? Well, consider the good old circle $x^2 + y^2 = C$. This works nicely when $C > 0$, but becomes a degenerate point when $C = 0$, and simply disappears for $C < 0$!

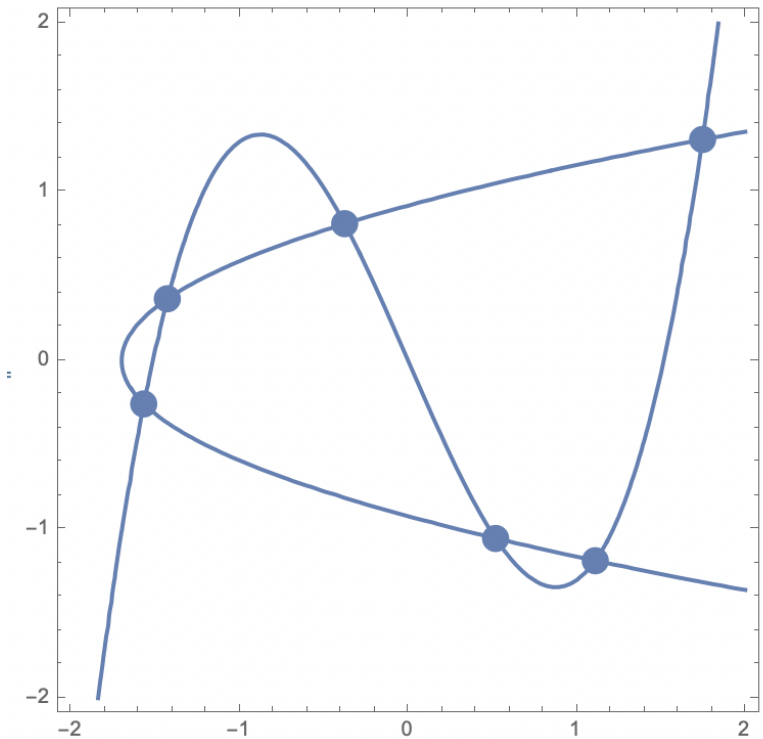
Using complex numbers solves this problem: All cases $C \neq 0$ becomes the same, and $C = 0$ at least becomes the union of two (complex) lines, instead of a single point (which has the wrong dimension: It's a point, not a curve!). To see this, notice that

$$x^2 + y^2 = (x + iy)(x - iy)$$

Unfortunately it's hard to draw pictures with 2 complex coordinates...

Bezout's theorem

A degree d_1 and a degree d_2 plane curve intersects in at most $d_1 d_2$ points in the plane. And it becomes exactly $d_1 d_2$ points, if understood in the complex projective plane, and counted with multiplicities!



Intersection of the curves $y = x^3 - 2.3x$ and $x = 2y^2 - 1.7$

Cubic curves

So after the lines (degree 1 curves) and conic sections (degree 2 curves), the next case is cubics (degree 3 curves). A typical cubic plane curve equation can look for example like this:

$$y^2 = x^3 + Ax + B$$

Unlike the previous cases, cubics are already complicated enough that: 1) they are *really interesting*; 2) there are very hard unsolved problems concerning them!

For example, solving the Birch and Swinnerton-Dyer conjecture would get you $1,000,000. Or the famous proof of Fermat's conjecture by Wiles was also (not so) secretly a statement about elliptic (cubic) curves.

Elliptic curves

So what is an *elliptic curve*?

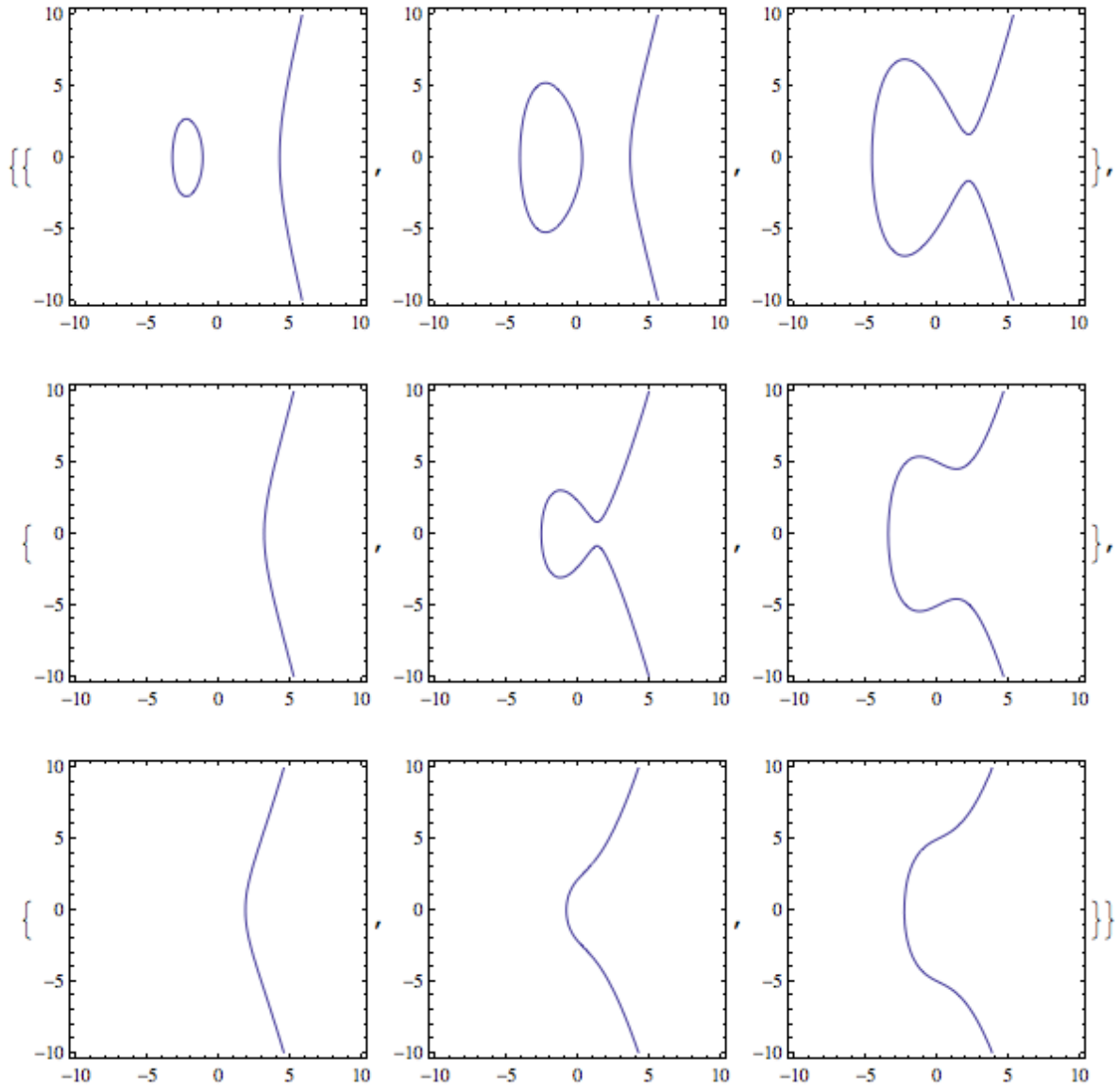
- a smooth cubic plane curve (together with a base point)
- smooth genus 1 curve (with a base point). Genus 1 means it has "1 hole", that is, it looks like a torus.
- a plane curve of the form $y^2 = x^3 + Ax + B$ with $\mathrm{char}(\mathbb{F}) \neq 2, 3$ and $4A^3 + 27B^2 \neq 0$; (called "short Weierstrass form")
- 1 dimensional abelian variety
- ...

Etimology / history:

- arc length of an ellipse
- elliptic integrals
- the inverse problem: elliptic functions
- elliptic curves

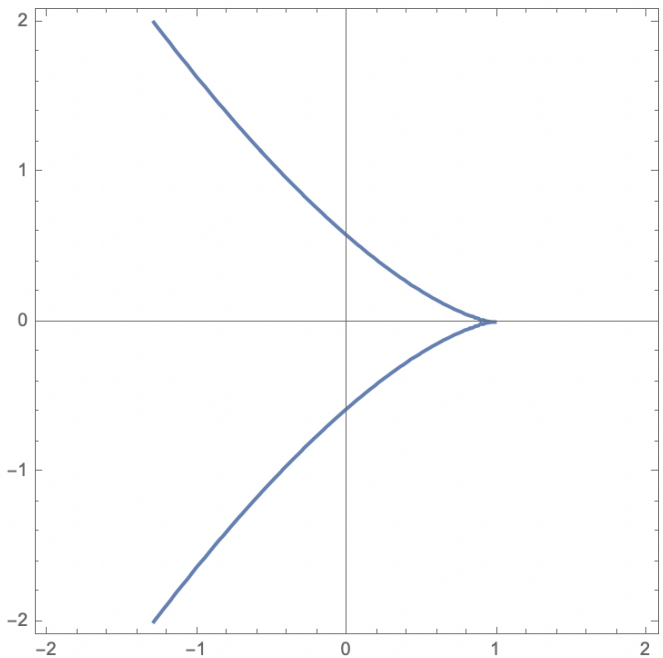
Pictures of elliptic curves over $\mathbb{R}$

```
Table[ContourPlot[y^2 == x^3 + a*x + b, {x, -10, 10}, {y, -10, 10}], {a, {-15, -5, +5}}, {b, {-15, +5, +25}}]
```

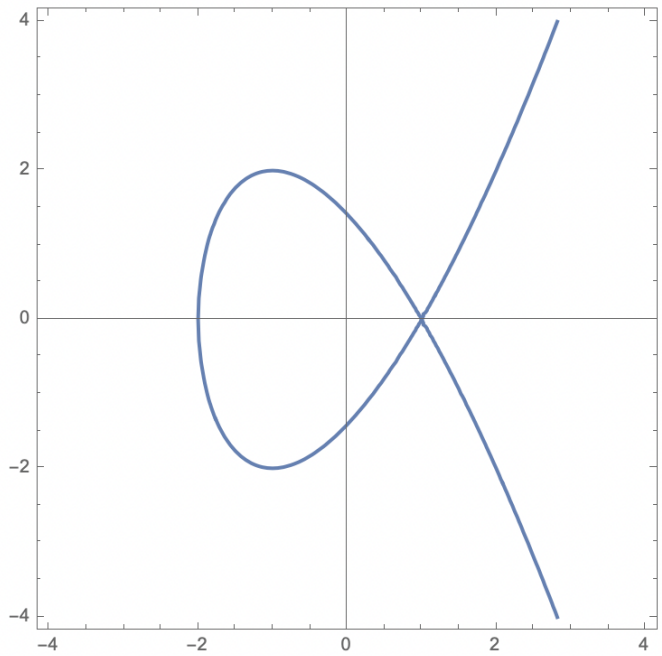


Singular cubics

Note that singular cubic curves are *NOT* called elliptic curves. There are two types of possible singularities for cubic curves, namely, double points and cusps:



Cusp



Double point

Brief history of the name "elliptic curve"

Why are we calling nonsingular cubic (pointed) curves "elliptic"?

- it all started with the arc length of the ellipse
- elliptic integrals (generalization of the arc length of the ellipse)
- elliptic functions (doubly periodic meromorphic functions)
- take the quotient wrt. the periodic lattice $\rightarrow$ (complex) elliptic curve groups

In the the next few slides, we will see this how the above 4 notions come together.

However, elliptic curves are a very central notion of contemporary mathematics, and they are related to many many other subjects too.

Arc length of an ellipse

An ellipse: $y = q\sqrt{1-x^2}$ with $q > 0$. The slope at (x, y):

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{-qx}{\sqrt{1-x^2}}$$

Let's write down the arc length:

$$S(t) = \int_0^t \sqrt{\mathrm{d}x^2 + \mathrm{d}y^2} = \int_0^t \sqrt{1 + (\mathrm{d}y/\mathrm{d}x)^2}\,\mathrm{d}x =$$
$$= \int_0^t \sqrt{\frac{1-x^2+q^2x^2}{1-x^2}}\,\mathrm{d}x =: E\big(t\,;\sqrt{1-q^2}\big)$$

For $q = 1$ (circle), this simplifies; otherwise it is not an elementary function (not even for $t = 1$). Remark:

$$E(t;k) = \int_0^t \sqrt{(1-k^2x^2)/(1-x^2)}\mathrm{d}x$$

is called "incomplete elliptic integral of the second kind, Jacobi's form".

Generic elliptic integral: $\int R(x, \sqrt{P(x)})\mathrm{d}x$, where $P(x)$ is a cubic or quartic polynomial without double roots, and $R(x, y)$ is a rational function.

Elliptic functions

We already know periodic functions like $\sin(x)$ and $\cos(x)$, and they are very interesting and useful.

What about *doubly periodic* functions? Let's fix two complex numbers $\omega_1, \omega_2 \in \mathbb{C}$; we are then looking for functions f for which $f(z) = f(z + \omega_1) = f(z + \omega_2)$. In other words, ω_1, ω_2 generates the *lattice*

$$\Lambda = \{ n\omega_1 + m\omega_2 \ : \ n, m \in \mathbb{Z} \} \subset \mathbb{C}.$$

and the function needs to be invariant wrt. translation to this lattice.

Unfortunately, all such (holomorphic) functions are just constants. However if we allow poles (infinity values at some points), then we can have interesting periodic "meromorphic" functions

Such doubly periodic meromorphic functions are called "elliptic functions".

Weierstrass $\wp$ function

Let us use $\Lambda_o = \Lambda \backslash \{0\}$ for brevity. The Weierstrass $\wp$ function is defined by

$$\wp(z) \;=\; \frac{1}{z^2} + \sum_{\omega \in \Lambda_o} \left[\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right]$$

(we need those funny "correction" terms to ensure that the sum converges).

It is meromorphic function (has double poles exactly at the points of Λ), periodic wrt. the lattice Λ (thus an elliptic function), and is an *even* function: $\wp(z) = \wp(-z)$.

It's not so clear that it is periodic, however the derivative $\wp'(z)$ is clearly periodic; and together with the evenness it follows that $\wp(z)$ is periodic too.

Fact: $\wp(z)$ is the *universal* elliptic function (wrt. Λ): Any elliptic function is a rational function of $\wp(z)$ and $\wp'(z)$.

The differential equation

Introduce the quantities:

$$g_2 = g_2(\Lambda) = \quad 60 \cdot \sum_{\omega \in \Lambda_o} \omega^{-4}$$

$$g_3 = g_3(\Lambda) = 140 \cdot \sum_{\omega \in \Lambda_o} \omega^{-6}$$

The Laurent series expansion of $\wp(z)$ can then be written as:

$$\wp(z) = z^{-2} + \frac{g_2}{20} z^2 + \frac{g_3}{28} z^4 + O(z^6)$$

Theorem: The Weierstrass $\wp$ function satisfies the following differential equation:

$$[\wp'(z)]^2 = 4\,\wp(z)^3 - g_2\,\wp(z) - g_3$$

Proof: Comparing the poles of the two sides, we can conclude that their difference is a periodic entire function, and thus a constant (which can be readily computed as 0).

The inverse of $\wp(z)$

The above differential equation looks quite interesting. So let's just play around with it!

Consider the derivative rule for inverse functions:

$$[f^{-1}]'(x) = \frac{1}{f'(f^{-1}(x))}$$

Applying this for $f = \wp$, we get another (simple) differential equation for the derivitive $[\wp^{-1}]'$

$$\begin{aligned}
[\wp^{-1}]'(y) &= \frac{1}{\wp'(\wp^{-1}(y))} = \\
&= \frac{1}{\sqrt{4\,\wp(\wp^{-1}(y))^3 - g_2\,\wp(\wp^{-1}(y)) - g_3}} = \\
&= \frac{1}{\sqrt{4y^3 - g_2 y - g_3}}
\end{aligned}$$

Conclusion I. (inverse problem)

Integrating the above differential equation, we can see that for the elliptic integral

$$u(y) = -\int_y^\infty \frac{1}{\sqrt{4s^3 - g_2 s - g_3}} \mathrm{d}s$$

we have $y = \wp(u(y))$, thus

$$\wp = u^{-1}.$$

That is, the Weierstrass $\wp$ function is the inverse of this elliptic integral. The "proof" is, again:

$$(\wp^{-1})'(y) = \frac{1}{\wp'(\wp^{-1}(y))} = \frac{1}{\sqrt{4y^3 - g_2 y - g_3}}$$

Similarly, other elliptic functions solve the inverse problems for other types of elliptic integrals (hence the name).

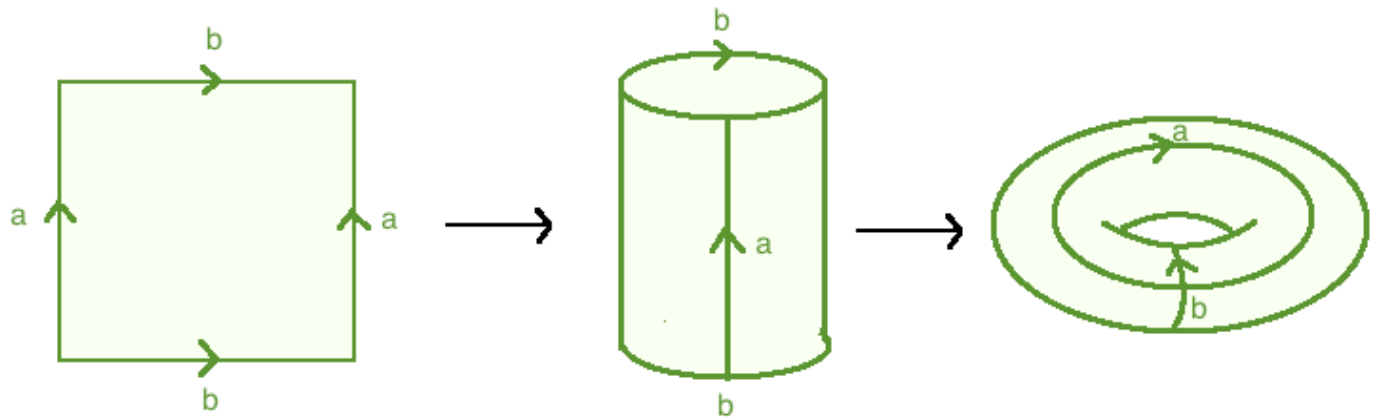
Complex tori

So an elliptic function is a complex function F (with poles allowed) which is periodic wrt. a lattice $\Lambda \subset \mathbb{C}$. In other words, for any $\lambda = n\omega_1 + m\omega_2 \in \Lambda$:

$$F(z) = F(z + \lambda) = F(z + n\omega_1 + m\omega_2)$$

What this means that F is really defined on the quotient $\mathbb{C}/\Lambda$!

This quotient is secretly a torus, and it also "inherits" the complex structure of the complex plane $\mathbb{C}$:



(image source: `http://mathonline.wikidot.com`)

Conclusion II. (elliptic curve)

From the differential equations, we can see directly that the mapping

$$\begin{array}{ccccc} \mathbb{C} & \to & \mathbb{C}/\Lambda & \to & \mathbb{P}^2 \\ z & \mapsto & z \bmod \Lambda & \mapsto & \left[4\wp(z) : 4\wp'(z) : 1\right] \end{array}$$

is well-defined (actually, an isomorphism) between the complex torus $\mathbb{C}/\Lambda$ and the elliptic curve

$$y^2 = x^3 + Ax + B$$

with $A = -4\, g_2(\Lambda)$ and $B = -16\, g_3(\Lambda)$.

So, *elliptic functions* are functions on the *elliptic curve*!

Moreover, $\Lambda \mapsto (g_2, g_3)$ is an isomorphism between the moduli space of lattices and the moduli space of elliptic curves (whatever that means...).

The group law on elliptic curves

The torus $\mathbb{C}/\Lambda$ is naturally a group (it inherits the complex addition), so it shouldn't be too surprising that elliptic curves also have a group structure. Actually it *is* rather surprising :)

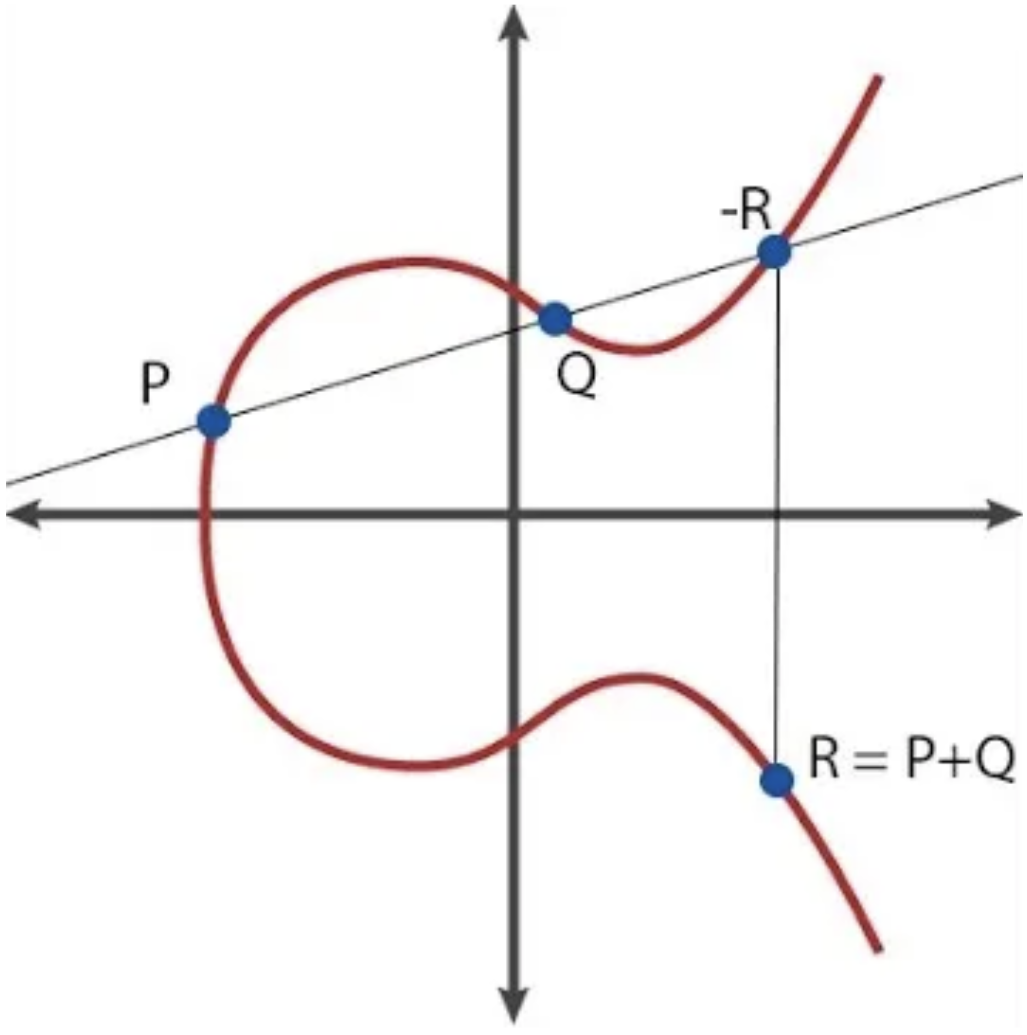
Definitions (for the Weierstrass form):

- identity element: The special point at infinity (denoted by O)
- inverse: mirroring wrt. the X axis
- addition: if P, Q and R are on a straight line, we declare $P + Q + R = O$

Group laws:

- identity satisfies what it should (trivial)
- addition is commutative (trivial)
- addition is associative (nontrival!)

Addition on elliptic curves



(image source: Vitalik Buterin)

Types of elliptic curves

Unlike conic sections, there are infinitely many different "types" of cubics, and elliptic curves. By 'different' here we mean that they are not isomorphic (cannot be changed into each other by coordinate transformations).

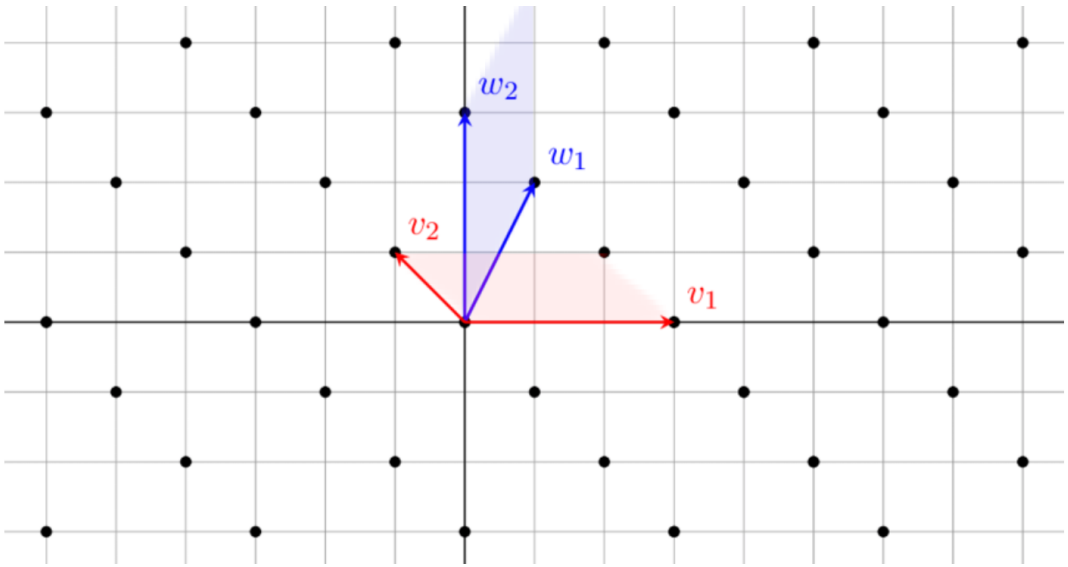
We have seen above that complex tori $\mathbb{C}^2/\Lambda$ are elliptic curves, and in fact the converse is also true: All (complex) elliptic curves are of the form $\mathbb{C}^2/\Lambda$. So we could in theory understand elliptic curves by understanding lattices $\Lambda \subset \mathbb{C}^2$ (and that feels quite a bit simpler!)

This will again result in more beautiful, interesting and important mathematics!

Space of lattices in $\mathbb{C}$

A 2D lattice is generated by two vectors (here complex numbers) $\omega_1, \omega_2 \in \mathbb{C}$. Because we only care about lattices up to rotation and scaling (which is the same as multiplication by a complex number), we can transform one of them into say $1 \in \mathbb{C}$; this way we get the pair $(\tau, 1)$ with $\tau = \omega_1/\omega_2 \in \mathbb{C}$. We can also assume wlog. that $\mathrm{im}(\tau) > 0$ (otherwise just mirror it).

However, different bases (or τ-s) can result in the same lattice. When does this happen?



$$w_1 = v_1 + 2v_2$$
$$w_2 = v_1 + 3v_2$$

Different basises of a lattice

Two pairs of vectors (v_1, v_2) and (w_1, w_2) generate the same lattice, iff. there are integers $a, b, c, d \in \mathbb{Z}$ such that

$$w_1 = av_1 + bv_2$$
$$w_2 = cv_1 + dv_2$$
$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc = \pm 1$$

The last condition here means the area of the spanned parallelogram remains the same; which in turns means that we don't "leave out" any lattice points.

It's easy to rewrite all this in the $\tau = \omega_1/\omega_2$ convention: It means that τ is mapped to

$$\tau \mapsto \frac{a\tau + b}{c\tau + d}$$

The groups $\mathrm{SL}_2(\mathbb{Z})$ and $\Gamma = \mathrm{PSL}_2(\mathbb{Z})$

To summarize:

- lattices in $\mathbb{C}$ (up to scaling and rotation) can be described by complex numbers $\tau \in \mathbb{H} = \{z \in \mathbb{C} \,:\, \mathrm{im}(z) > 0\}$ ($\mathbb{H} \subset \mathbb{C}$ is called the 'upper halfplane')
- two lattices defined by τ and τ' are the same if $\tau' = (a\tau + b)/(c\tau + d)$ for some integers $a, b, c, d \in \mathbb{Z}$ with $ad - bc = 1$ (we don't need the -1 option, that would mean $\mathrm{im}(\tau') < 0$)

We can rephrase all this that group $\mathrm{SL}_2(\mathbb{Z})$ (group of 2×2 integral matrices with determinant 1) acts on the upper halfplane $\mathbb{H}$ by $\tau \mapsto (a\tau + b)/(c\tau + d)$, and the quotient $\mathbb{H}/\mathrm{SL}_2(\mathbb{Z})$ is the "space of lattices" in $\mathbb{C}$ up to scaling and rotation.

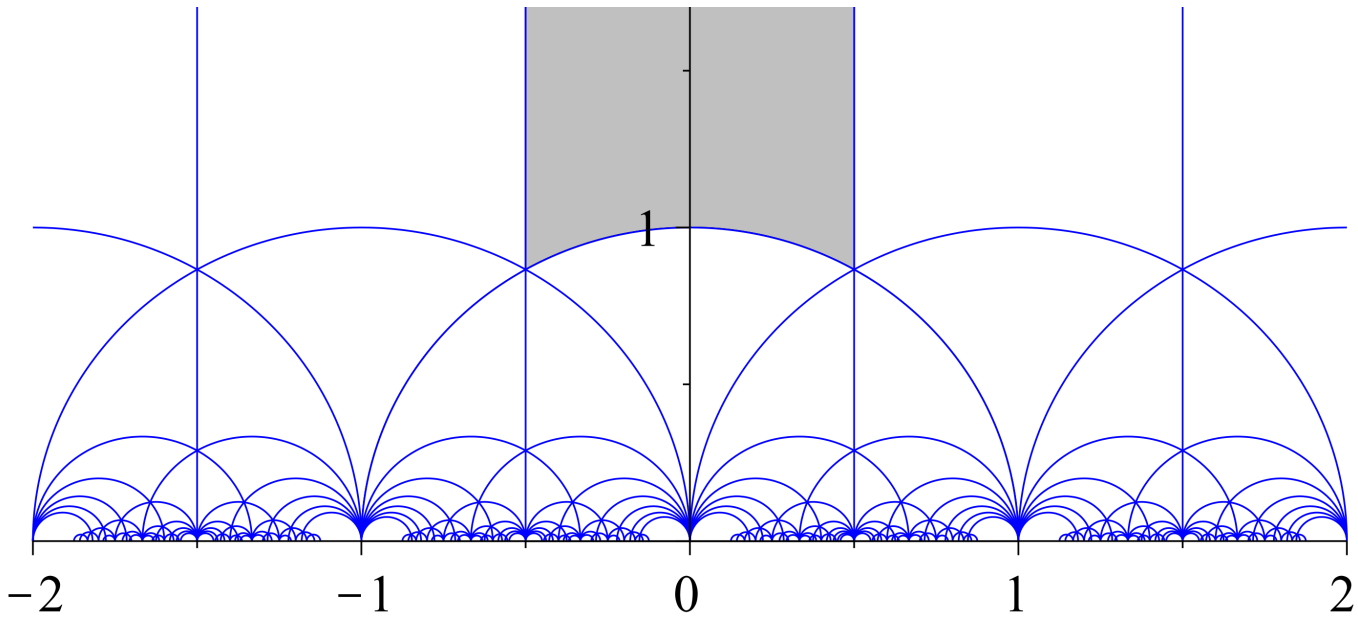
Note: negating all four of a, b, c, d leaves τ the same, so we don't really need that; it's better to use $\mathrm{PSL}_2(\mathbb{Z}) = \mathrm{SL}_2(\mathbb{Z})/\{+1, -1\}$.

The fundamental domain

The following two matrices in fact generate $\mathrm{PSL}_2(\mathbb{Z})$:

$$S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \quad \tau \mapsto -\frac{1}{\tau} \qquad\qquad T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \quad \tau \mapsto \tau + 1$$

Using this fact, one can draw the following picture:



(image source: wikipedia)

The j-invariant

The j-invariant is a complex function which can distinguish elliptic curves from each other.

One way to write it is quite simple: For the curve $\mathcal{C}$ defined by the equation $y^2 = x^3 + Ax + B$, we have $j(\mathcal{C}) = 1728 \times 4A^3/(4A^3 + 27B^2)$. This looks deceivingly simple, but the truth is way more interesting!

We can also write it terms of $\tau \in \mathbb{H}$. By definition, it must be invariant to the action of $\mathrm{SL}_2(\mathbb{Z})$, and that already makes it something very interesting, called a "modular function"!

Plot of $j(\tau)$ (image source: Fredrik Johansson)

Some formulas for the j-invariant

$$j(\tau) = 1728\,\frac{g_2(\tau)^3}{g_2(\tau)^3 - 27g_3(\tau)^2} = 1728\,\frac{g_2(\tau)^3}{\Delta(\tau)} = \left[12\frac{g_2(\tau)}{(2\pi)^4\eta(\tau)^8}\right]^3$$

where recall that

$$g_2(\tau) = 60\sum_{(m,n)\neq(0,0)}(m+n\tau)^{-4} \qquad g_3(\tau) = 140\sum_{(m,n)\neq(0,0)}(m+n\tau)^{-6}$$

$\Delta(\tau) = g_2(\tau)^3 - 27g_3(\tau)^2 = (2\pi)^{12}\eta(\tau)^{24}$ is called the "modular discriminant"; and η is called Dedekind eta, and is defined by (with $q = e^{2\pi i\tau}$)

$$\eta(\tau) = q^{1/24}\prod_{n=1}^{\infty}(1-q^n)$$

There are many many other fascinating formulas and other stuff related to these, check out the wikipedia page for a quick look!

Monstrous moonshine

The Monster group is the largest finite sporadic simple group, it has approx. 8×10^{53} elements, more precisely:

$$|M| = 808,017,424,794,512,875,886,459,904,961,710,757,005,754,368,000,000,000$$

Now compare the q-expansion (here $q = e^{2\pi i \tau}$) of the j-invariant:

$$j(\tau) = q^{-1} + 196884q + 21493760q^2 + 864299970q^3 + 20245856256q^4 + \dots$$

with the dimensions $r_1, r_2, r_3, \dots$ of the smallest irreducible representations of the Monster group:

$$\begin{aligned}
1 &= r_1 \\
196884 &= r_1 + r_2 = 1 + 196883 \\
21493760 &= r_1 + r_2 + r_3 \\
864299970 &= 2r_1 + 2r_2 + r_3 + r_4 \\
20245856256 &= 3r_1 + 3r_2 + r_3 + 2r_4 + r_5 = 2r_1 + 3r_2 + 2r_3 + r_4 + r_6 \\
\dots &= \dots
\end{aligned}$$

Yeah, like, WTF?! And that's exactly what every mathematician thought when this was first discovered!

Elliptic curves over finite fields

Elliptic curves make sense over different fields: $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{F}_q$, function fields, etc. So far we were looking at $\mathbb{R}$ and $\mathbb{C}$, but the other cases are interesting too (in fact, using the viewpoint of modern mathematics, you should really try and look at all of them at the same time!).

For example elliptic curves over both $\mathbb{Q}$ and finite fields $\mathbb{F}_q$ are of great interest for number theorists, and the finite field version also for cryptographers.

Among the many many interesting questions, we can for example ask how many points an elliptic curve has over a finite field $\mathbb{F}_q$ (there are finitely many possible coordinates, so this will be always a finite number).

Counting points

Given a curve equation $y^2 = x^3 + Ax + B$ of a curve E with $A, B \in \mathbb{Z}$, we can make it sense in any field (or ring), since there is a unique morphism from $\mathbb{Z}$ to any ring.

We can then ask how the number of points $|E(\mathbb{F}_q)|$ over the field of size q varies when as we change q. Recall that finite fields exist only for $q = p^k$ where p is a prime; nevertheless for any such q we can form the series

$$N_m := |E(\mathbb{F}_{q^m})|$$

The "local zeta function" is then defined as

$$Z_{E/\mathbb{F}_q}(t) := \exp\left(\sum_{m=1}^{\infty} N_m \frac{t^m}{m}\right) = \frac{1 - (q + 1 - |E(\mathbb{F}_q)|) \cdot t + qt^2}{(1-t)(1-qt)}$$

and the "global Hasse-Weil zeta function" is (almost...) the infinite product over all primes p of

$$\zeta_E(s) \approx \prod_p Z_{E/\mathbb{F}_p}(p^{-s})$$

This is closely related to the famous Riemann zeta function!

Elliptic curves in cryptography

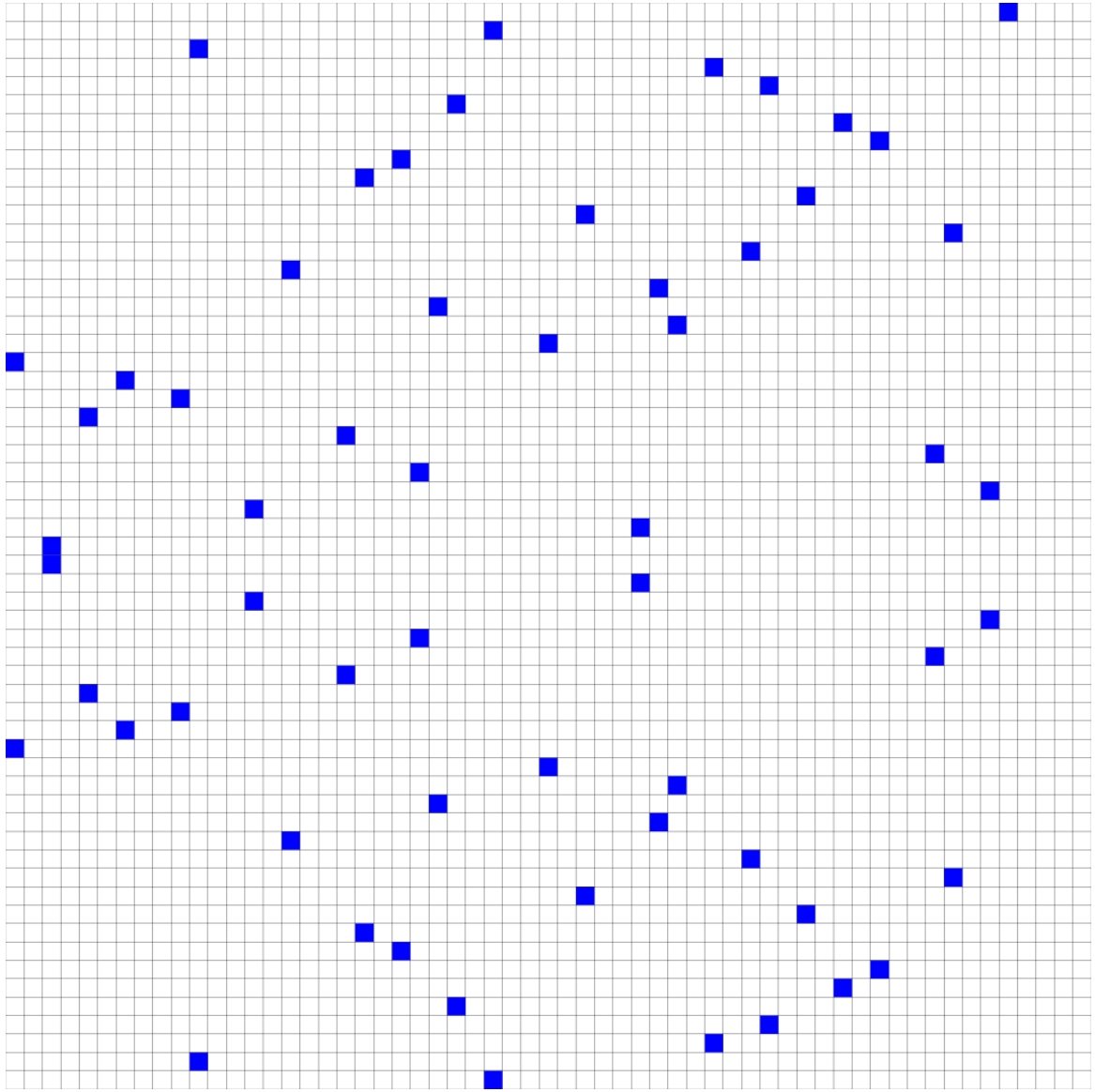
To be able to do cryptography, we need computable objects; thus the natural choice of finite fields $\mathbb{F}_q$. In practice either $q = 2^m$ or a prime $q = p$ (exception: pairing-based cryptography).

Furthermore, the safety of ECC depends on the (conjectured) hardness of the elliptic curve discrete logarithm problem. Note that not all curves are created equal in this sense!

An example, here is the standardized curve called `secp256k1` (this is the curve used by Bitcoin):

- the field is $\mathbb{F}_p$ with $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1 \approx 10^{77}$
- the curve equation is $y^2 = x^3 + 7$ (that is, $A = 0$ and $B = 7$)
- the number of points on the curve n is also a prime, and "close" to p (to be more precise, $p - 2^{129} < n < p - 2^{128}$)
- since n is prime, the group is cyclic; thus any element (except the infinity) will do as the generator G
- but there is a concrete, randomly-looking G in the standard

A picture of $y^2 = x^3 + 7$ over $\mathbb{F}_{59}$



Remark: $p \approx 2^{256} \approx 10^{77} \approx$ no. of elementary particles in the universe

Symmetric key cryptography

The two parties have a shared secret key; they can then encrypt and decrypt messages using this key.

With modern symmetric key encryption standards, it is infeasible for an attacker to guess the message without knowing the secret key.

The big issue: *How to agree on a secret key?* In practice, this needs meeting in person (often inpractical) and at a secure location (can be inpractical, or even impossible). Also, we want machines to communicate safely, too.

Nevertheless, symmetric key encryption algorithms are useful components of larger crypto systems.

Asymmetric (or public) key cryptography

Each party has a *pair* of corresponding keys: one which is public (say, published on their homepage) and one which is private (only a single person knows it).

Main applications:

- Estabilishing a shared secret without meeting (key exchange)
- Send messages which only the intended recepient can decrypt (encryption)
- Prove that a message was really written by the person who claims it (signature)

From these basic building blocks, a huge set of really interesting applications can be built.

Public-key crypto is widely used on the internet today: HTTPS, SSL, PGP, Bitcoin, Ethereum, etc...

Public key cryptography, II.

Public-key cryptosystems are based on problems which are easy to compute in one direction, but hard to compute in the other direction:

- factorization of the product of two large prime numbers (RSA)
- discrete logarithm (ElGamal)
- elliptic curve discrete logarithm (ECC)

Discrete logarithm: Fix a finite cyclic group $\mathbb{G}$ of order n and a generator $\mathbf{g} \in \mathbb{G}$.

- private key: a random number $d \in [1, n-1] \subset \mathbb{N}$
- public key: the group element $Q = d * \mathbf{g} \in \mathbb{G}$

The idea is that it is very hard to determine d from Q (for appropriate choices of $\mathbb{G}$). Elliptic curve crypto: Let $\mathbb{G}$ be a subgroup of an appropriately chosen elliptic curve over a finite field.

Diffie-Hellmann key exchange

Recall that a *key pair* (d, Q) consists of

- a private key, which is a random number $d \in [1, n-1] \subset \mathbb{N}$
- a public key, which is the group element $Q = d * \mathbf{g} \in \mathbb{G}$

Let there be two parties: Alice and Bob, with key pairs (d_A, Q_A) and (d_B, Q_B). They can compute a shared secret $S \in \mathbb{G}$ as follows:

$$d_A * Q_B = d_A * (d_B * \mathbf{g}) = \underbrace{(d_A d_B) * \mathbf{g}}_{S} = d_B * (d_A * \mathbf{g}) = d_B * Q_A$$

Alice can knows d_A, so she can compute the leftmost version; Bob knows d_B, so he can compute the rightmost version. But nobody else knows neither d_A or d_B, thus S is their secret.

They can then proceed and use S for any purpose, for example as the key of a symmetric encryption scheme.

Public-key encryption

Alice wants to send a message to Bob, but wants to make sure that nobody else can read it.

This can be implemented as an application of the Diffie-Hellmann key exchange:

1. Alice generates an ephemeral key pair (d_E, Q_E)
2. She then computes a shared secret $S = d_E * Q_B$
3. computes a symmetric key $k = k(S)$ from S
4. encrypts the message m with the key k
5. sends Q_E and the ciphertext $c_k(m)$ to Bob

On the other side: Bob computes $S = d_B * Q_E$, then $k = k(S)$, and decrypts the message. Nobody else knows neither d_E or d_B.

In practice it is a bit more complicated, but that's the idea.

Elliptic Curve Digital Signature Algorithm

Alice writes a message m, and wants to prove that she wrote it. She already has key pair (d_A, Q_A), and people accept that the public key Q_A in fact belongs to her.

Construction of the signature:

1. compute a hash $z = \mathrm{HASH}(m) \in [1, n-1]$ of the message m
2. generate an ephemeral key pair: k and $Q_k = k * \mathbf{g} = (x, y)$
3. let $r = (x \bmod n) \in \mathbb{Z}_n$
4. let $s = k^{-1}(z + rd_A) \in \mathbb{Z}_n$
5. the signatures is $(r, s) \in \mathbb{Z}_n \times \mathbb{Z}_n$

Verification of the signature:

1. compute $z = \mathrm{HASH}(m) \in [1, n-1]$ as before
2. compute $u = s^{-1}z \in \mathbb{Z}_n$ and $v = s^{-1}r \in \mathbb{Z}_n$
3. compute the curve point $(x, y) = Q = u * \mathbf{g} + v * Q_A \in \mathbb{G}$
4. the signature is valid iff $x = r$.

Computations with elliptic curves, I.

How to compute $d * Q \in \mathbb{G}$ efficiently, with $d \in \mathbb{Z}_n$ and $Q \in \mathbb{G}$?

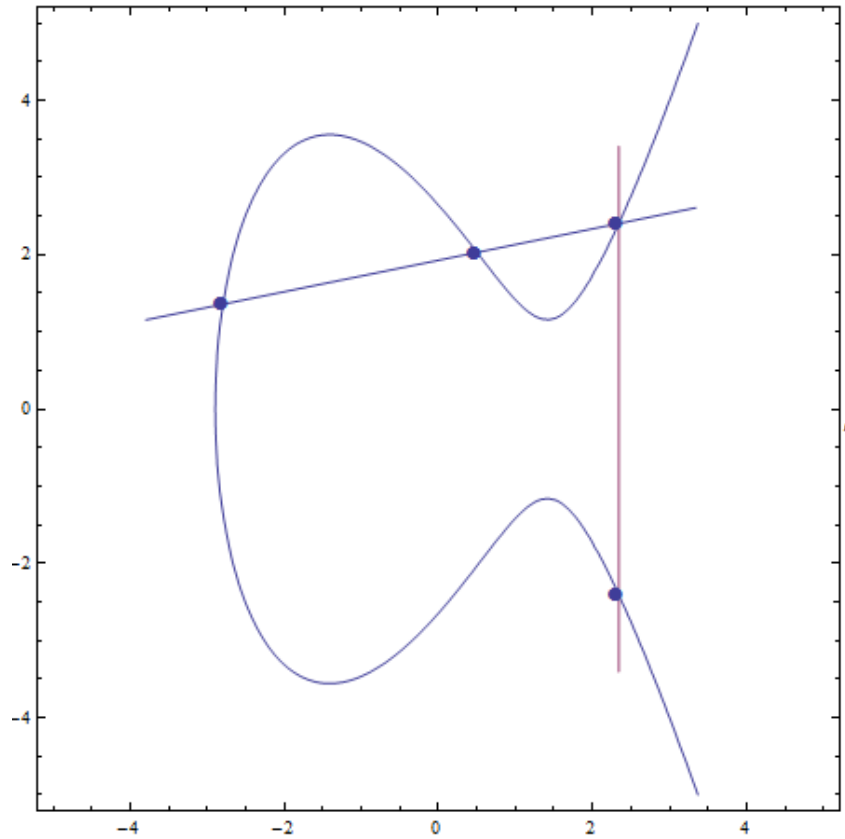
Answer: "fast exponentation"! Write d in binary form: $d = \sum_{i=0}^{m-1} d_i 2^i$, where $d_i \in \{0, 1\}$.

$$d * Q = \left(\sum_{i=0}^{m-1} d_i 2^i \right) * Q = \sum_{i=0}^{m-1} d_i * (2^i * Q) = \sum_{i=0}^{m-1} d_i * Q_i$$

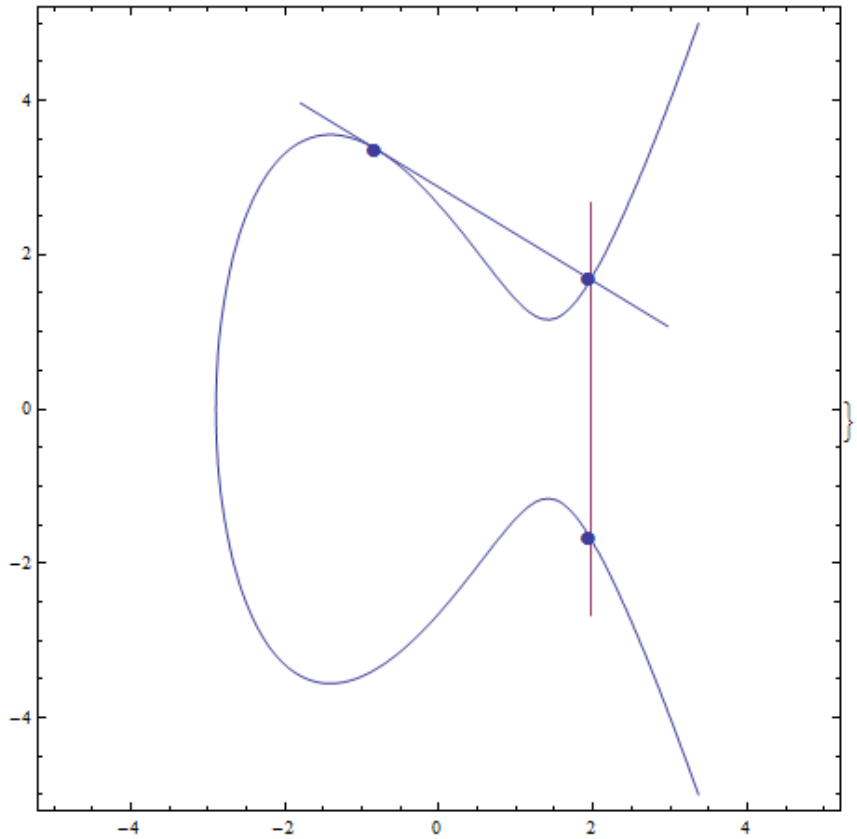
where $Q_i = 2^i * Q$ can be computated by *repeated doubling*: $Q_0 = Q$, $Q_1 = 2 * Q_0$, $Q_2 = 2 * Q_1$, $Q_3 = 2 * Q_2$, etc...

Thus we need addition and doubling (which is a special case of addition, but needs to be handled separately anyway).

Elliptic curve addition and doubling in pictures



Addition of distinct points



Doubling of a point

Elliptic curve addition and doubling in Weierstrass form

In any field $\mathbb{F}$ (at least with $\mathrm{char}(\mathbb{F}) \neq 2, 3$), for two distinct points $P \neq Q \neq O$ on the elliptic curve $y^2 = x^3 + ax + b$, with coordinates $P = (x_p, y_p)$ and $Q = (x_q, y_q)$, it is relatively straightforward to calculate the coordinates of $P + Q = R = (x_r, y_r)$ and $2P = U = (x_u, y_u)$:

$$
\begin{aligned}
s &= \frac{y_q - y_p}{x_q - x_p} & \qquad t &= \frac{3x_p^2 + a}{2y} \\
x_r &= s^2 - (x_p + x_q) & \qquad x_u &= t^2 - 2x_u \\
y_r &= -y_p - s(x_r - x_p) & \qquad y_u &= -y_p - s(x_u - x_p)
\end{aligned}
$$

Here s resp. t are the slopes of the secant resp. tangent lines.

It's possible to derive the formulas from either the geometric definition or via the $\wp(z)$ functions.

Projective coordinates

The problem: Each addition or doubling needs a division in $\mathbb{F}_q$, which is *slow*; each exponentiation needs a *lot* of additions and doublings; and we need several exponentiation to do cryptography.

The divisions are the bottleneck. How to make divisions faster? Answer: Don't do divisions!

Using projective coordinates, we only need one division at the end, when we convert back to affine coordinates. Using weighted projective coordinates $\mathbb{P}(2,3,1)$ can be even better (at least for some computations):

$$[x:y:z] = [\lambda^2 x : \lambda^3 y : \lambda z] \in \mathbb{P}(2,3,1).$$

Other representations can be even more efficient.

Pairings

"Pairings" is just another name for a bilinear map between groups:

$$\langle -, - \rangle : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$$

That is, it should have the following properties:

$$\begin{aligned} \langle g_1 + g_2, h \rangle &= \langle g_1, h \rangle \cdot \langle g_2, h \rangle \\ \langle g, h_1 + h_2 \rangle &= \langle g, h_1 \rangle \cdot \langle g, h_2 \rangle \\ \langle d * g, h \rangle = \langle g, d * h \rangle &= \langle g, h \rangle^d \end{aligned}$$

Note: for *reasons*, the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are usually written with additive notation, while $\mathbb{G}_t$ is written with multiplicative notation. The latter is because $\mathbb{G}_t$ is usually a subgroup of the multiplicative group of a finite field $\mathbb{F}_{p^k}$.

Pairing-based cryptography

With really careful choices, we can have such pairings with an elliptic curve E, with practically useful properties.

$$\begin{aligned} \mathbb{G}_1 &\subset E(\mathbb{F}_p) \\ \mathbb{G}_2 &\subset E(\mathbb{F}_{p^k}) \\ \mathbb{G}_2 \cong \mathbb{G}_2' &\subset E(\mathbb{F}_{p^2}) \\ \mathbb{G}_t &\subset \mathbb{F}_{p^k}^{\times} \end{aligned}$$

Note: $k \in \mathbb{N}$ (called the "embedding degree") depends on the elliptic curve E, and is a really dedicated balancing act (too small means insecure; too big means too inefficient). In practice we usually have $k = 12$.

Let's just take the above on faith, it's way more complicated than most of stuff so far...

BLS signatures

An example of pairing-based cryptography is BLS digital signature:

- private key: $d \in \mathbb{N}$
- public key: $Q := d * \mathbf{g}_1 \in \mathbb{G}_1$
- message m; hash it into the curve $h = \mathrm{HASH}(m) \in \mathbb{G}_2$
- signature: $\sigma := d * h \in \mathbb{G}_2$
- verification: check that $\langle \mathbf{g}_1, \sigma \rangle = \langle Q, h \rangle$

Why does this work?

$$\langle \mathbf{g}_1, \sigma \rangle = \langle \mathbf{g}_1, d * h \rangle = \langle \mathbf{g}_1, h \rangle^d = \langle d * \mathbf{g}_1, h \rangle = \langle Q, h \rangle$$

Note: the role of $\mathbb{G}_1$ and $\mathbb{G}_2$ is interchangeable here.

Pairings are useful because they allow to check a "single multiplication". Even more spectacular usage of pairings can be found in (zk-)SNARKS (zero-knowledge proofs).