Computing two-dimensional isogenies for SQIsign

Maria Corte-Real Santos
CNRS and ENS de Lyon

I. Motivation

A new tool

Interpolation

Given coprime integers $N < N'$ and four points P_1, P_2, Q_1, Q_2 such that $\langle P_1, P_2 \rangle = E_1[N']$ and $\langle Q_1, Q_2 \rangle = E_2[N']$ with N' a B-powersmooth integer.

One can **efficiently check existence** and **compute** the isogeny of degree N

$$\varphi : E_1 \to E_2 \text{ such that } \varphi(P_1) = Q_1 \text{ and } \varphi(P_2) = Q_2.$$

Damien Robert. *On the efficient representation of isogenies* (2024).

A new tool

Interpolation

Given coprime integers $N < N'$ and four points P_1, P_2, Q_1, Q_2 such that $\langle P_1, P_2 \rangle = E_1[N']$ and $\langle Q_1, Q_2 \rangle = E_2[N']$ with N' a B-powersmooth integer.

One can **efficiently check existence** and **compute** the isogeny of degree N

$\varphi : E_1 \to E_2$ such that $\varphi(P_1) = Q_1$ and $\varphi(P_2) = Q_2$.

Requires computing dimension ≥ 2 isogenies of degree ℓ between products of elliptic curves.

Damien Robert. *On the efficient representation of isogenies* (2024).

SQIsign

- New variants of SQIsign exploit this new tool.

SQIsign

- New variants of SQIsign exploit this new tool.

- For a sufficiently nice set-up, we can restrict to computing 2D isogenies.

SQIsign

- New variants of SQIsign exploit this new tool.
- For a sufficiently nice set-up, we can restrict to computing 2D isogenies.

SQIsign2D–West

The Fast, the Small, and the Safer

Andrea Basso[1,2][0000–0002–3270–1069], Pierrick Dartois[3,4][0009–0008–2808–9867], Luca De Feo[2][0000–0002–9321–0773], Antonin Leroux[5,6][0009–0002–3737–0075], Luciano Maino[1][0009–0005–4495–5102], Giacomo Pope[1,7], Damien Robert[3,4][0000–0003–4378–4274], and Benjamin Wesolowski[8][0000–0003–1249–6077]

SQIsign2D-East: A New Signature Scheme Using 2-dimensional Isogenies

Kohei Nakagawa[1] and Hiroshi Onuki[2]

SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies

Max Duparc and Tako Boris Fouotsa

SQIsign

- New variants of SQIsign exploit this new tool.

- For a sufficiently nice set-up, we can restrict to computing 2D isogenies.

- Two-dimensional isogenies are used in key generation, signing *and* verification!

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Degree ℓ^k

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Degree ℓ^k

$R, S \in E_1 \times E_2[\ell^k]$ generating kernel

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Degree ℓ^k

$R, S \in E_1 \times E_2[\ell^k]$ generating kernel

$$\xrightarrow[\Phi]{}$$

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Degree ℓ^k

$R, S \in E_1 \times E_2[\ell^k]$ generating kernel

$\xrightarrow{\Phi}$

Outputs

Codomain $E_3 \times E_4$

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Degree ℓ^k

$R, S \in E_1 \times E_2[\ell^k]$ generating kernel

Points $P_1, \ldots, P_n \in E_1 \times E_2$

$$\xrightarrow{\Phi}$$

Outputs

Codomain $E_3 \times E_4$

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Primes $p \neq \ell$

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Degree ℓ^k

$R, S \in E_1 \times E_2[\ell^k]$ generating kernel

Points $P_1, \ldots, P_n \in E_1 \times E_2$

$\xrightarrow{\Phi}$

Outputs

Codomain $E_3 \times E_4$

Image points $\Phi(P_1), \ldots, \Phi(P_n)$

Our goal

$$\Phi : E_1 \times E_2 \longrightarrow E_3 \times E_4$$

Inputs

Cryptographic sized p

Domain $E_1 \times E_2$ defined over $\bar{\mathbb{F}}_p$

Degree 2^k

$R, S \in E_1 \times E_2[2^k]$ generating kernel

Points $P_1, \ldots, P_n \in E_1 \times E_2$

$\xrightarrow{\Phi}$

Outputs

Codomain $E_3 \times E_4$

Image points $\Phi(P_1), \ldots, \Phi(P_n)$

Structure of Talk

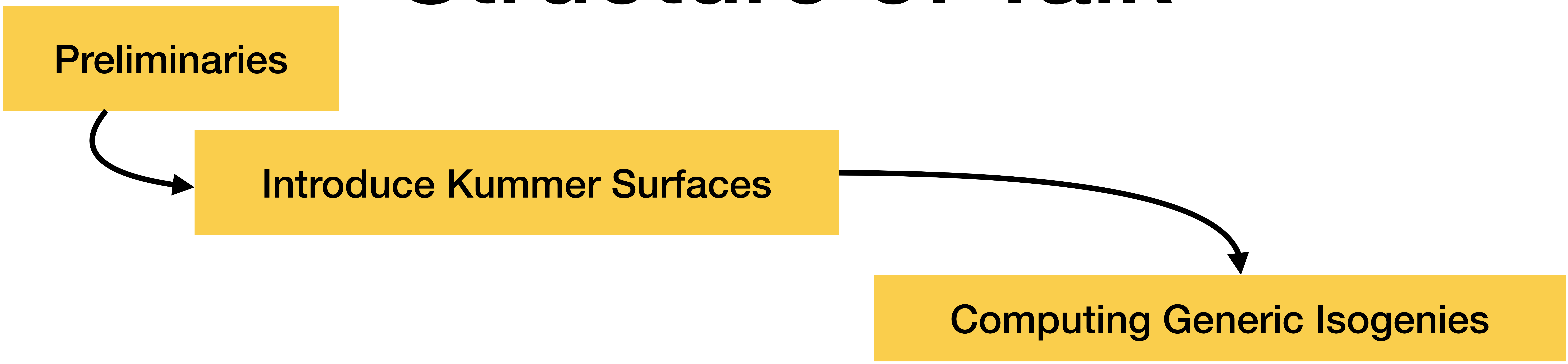
Preliminaries

Structure of Talk

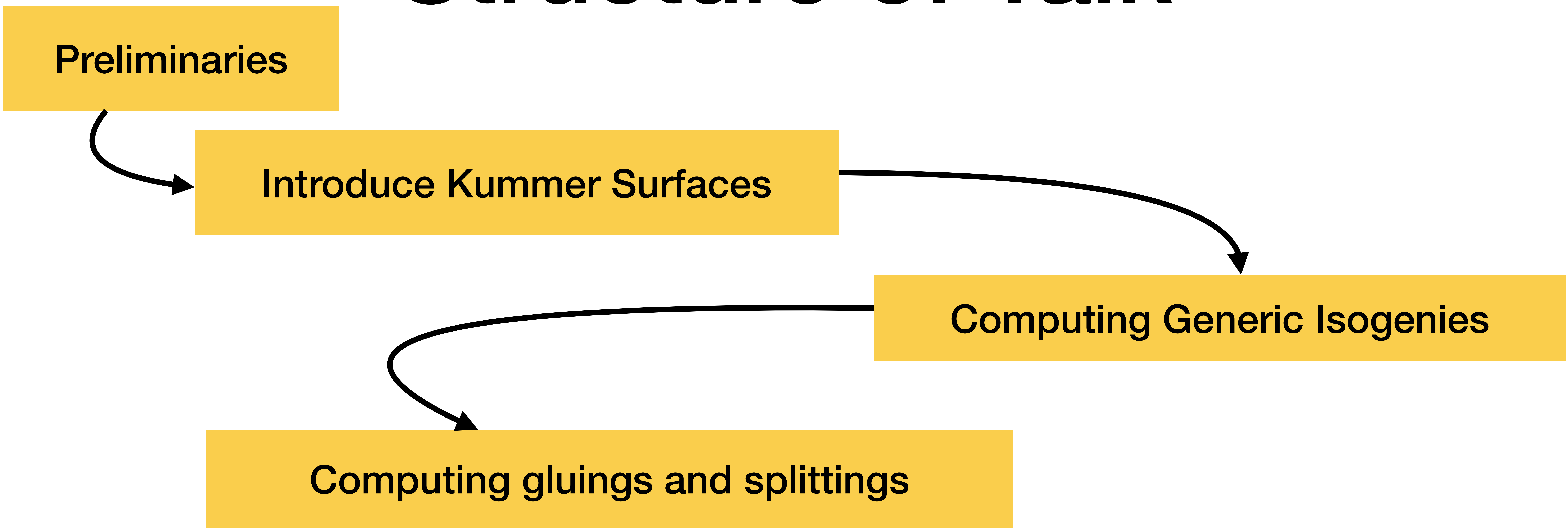
Preliminaries

Introduce Kummer Surfaces

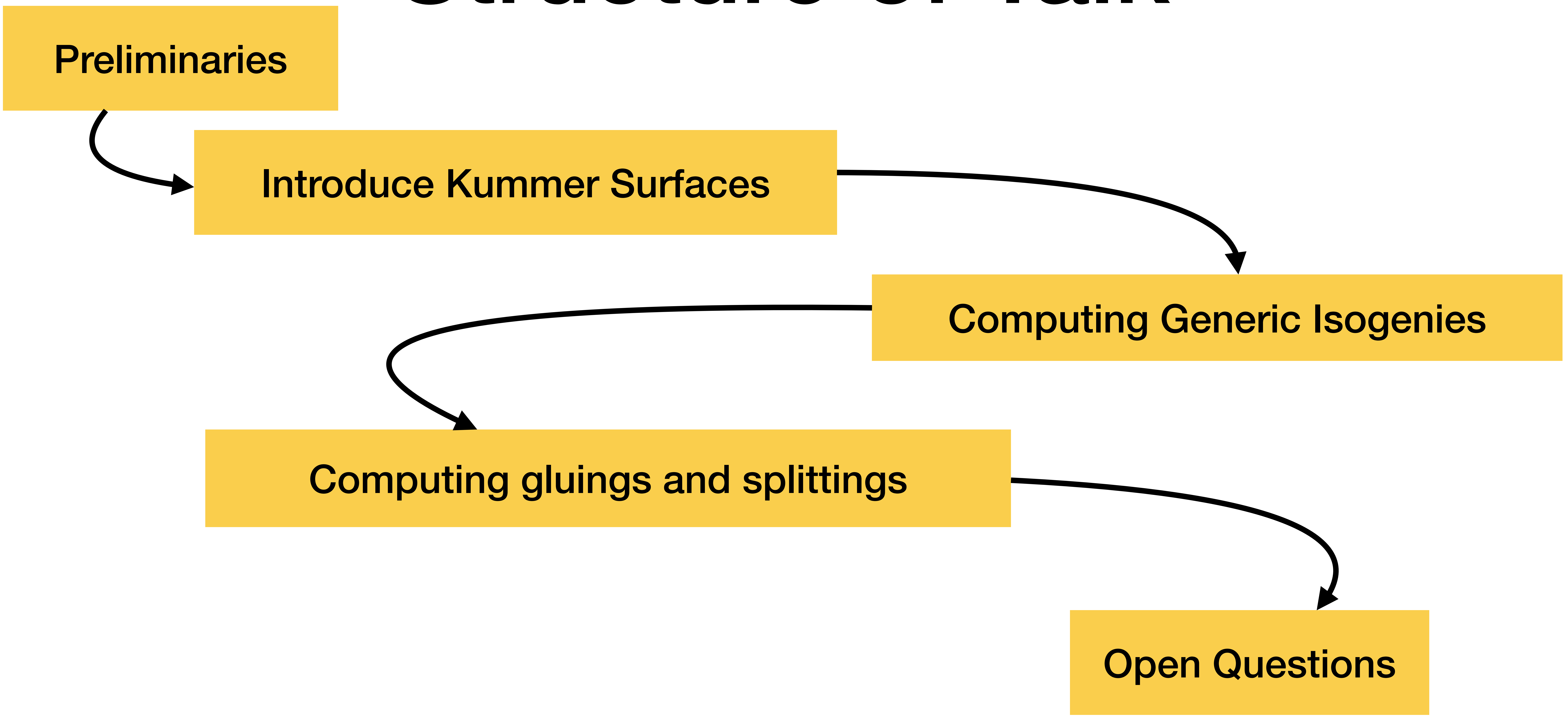
Structure of Talk



Structure of Talk



Structure of Talk



II. Preliminaries

Moving to higher dimension

We now consider **superspecial principally polarised abelian surfaces** (A, λ).

Moving to higher dimension

We now consider **superspecial principally polarised abelian surfaces** (A, λ).

gives you an embedding
into projective space

Moving to higher dimension

We now consider **superspecial** **(principally polarised)** **abelian surfaces** (A, λ).

Moving to higher dimension

We now consider **superspecial** **(principally polarised)** **abelian surfaces** (A, λ).

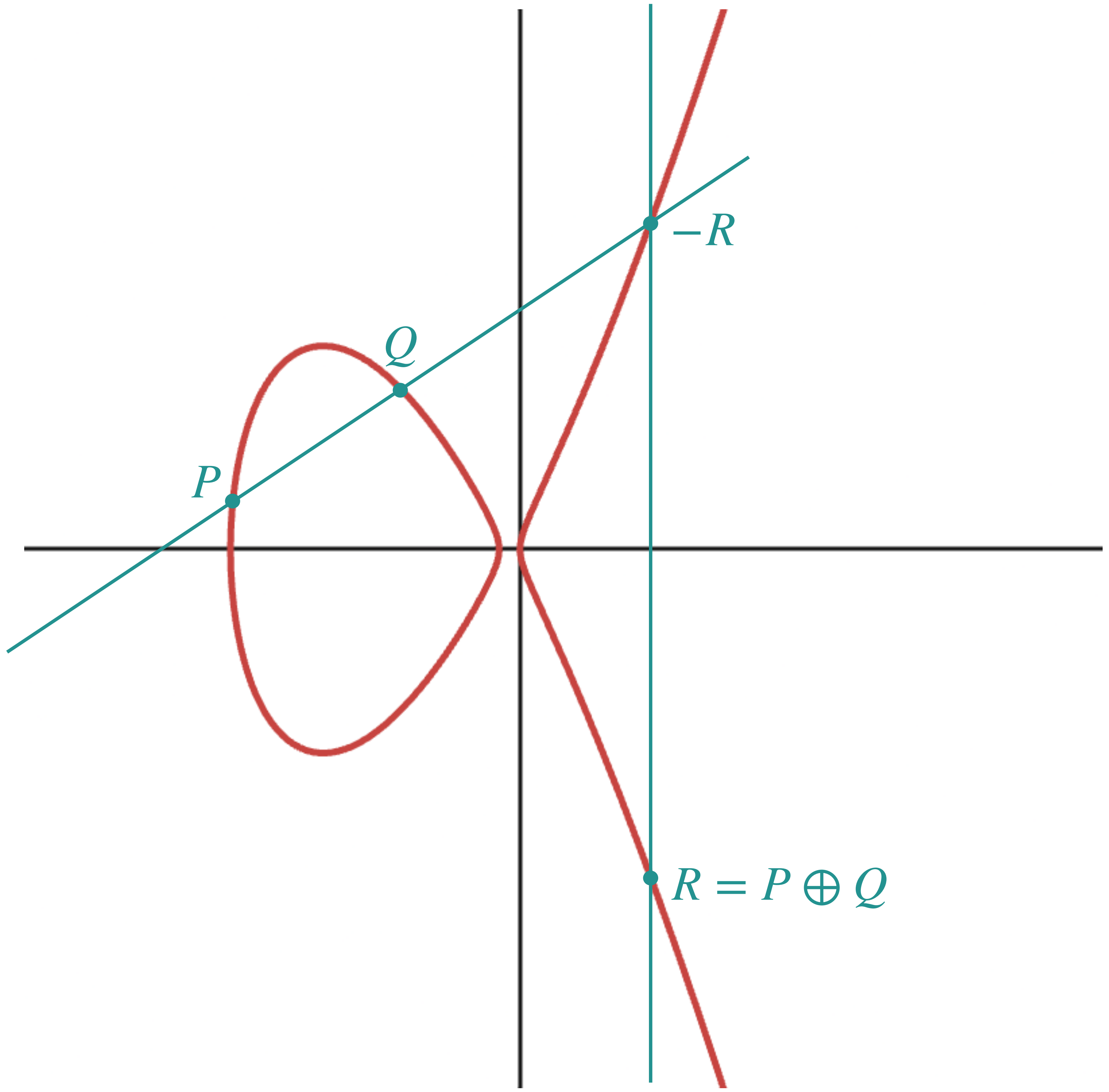
There are two types:

- Products of elliptic curves $E_1 \times E_2$
- Jacobians J_C of genus 2 hyperelliptic curves C

Moving to higher dimension

Elliptic Curves

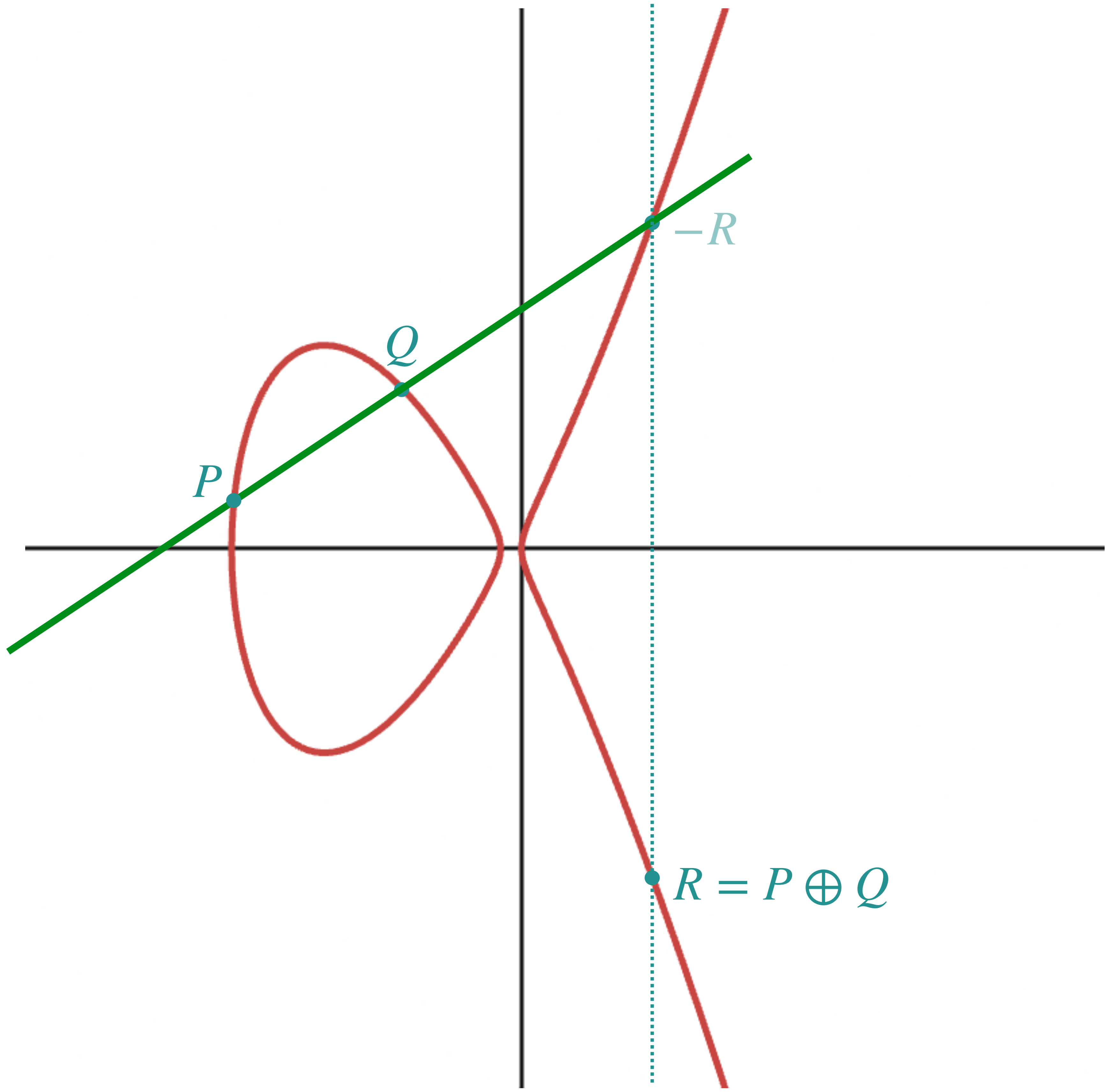
$$E\colon y^2 = x^3 + Ax + B$$



Moving to higher dimension

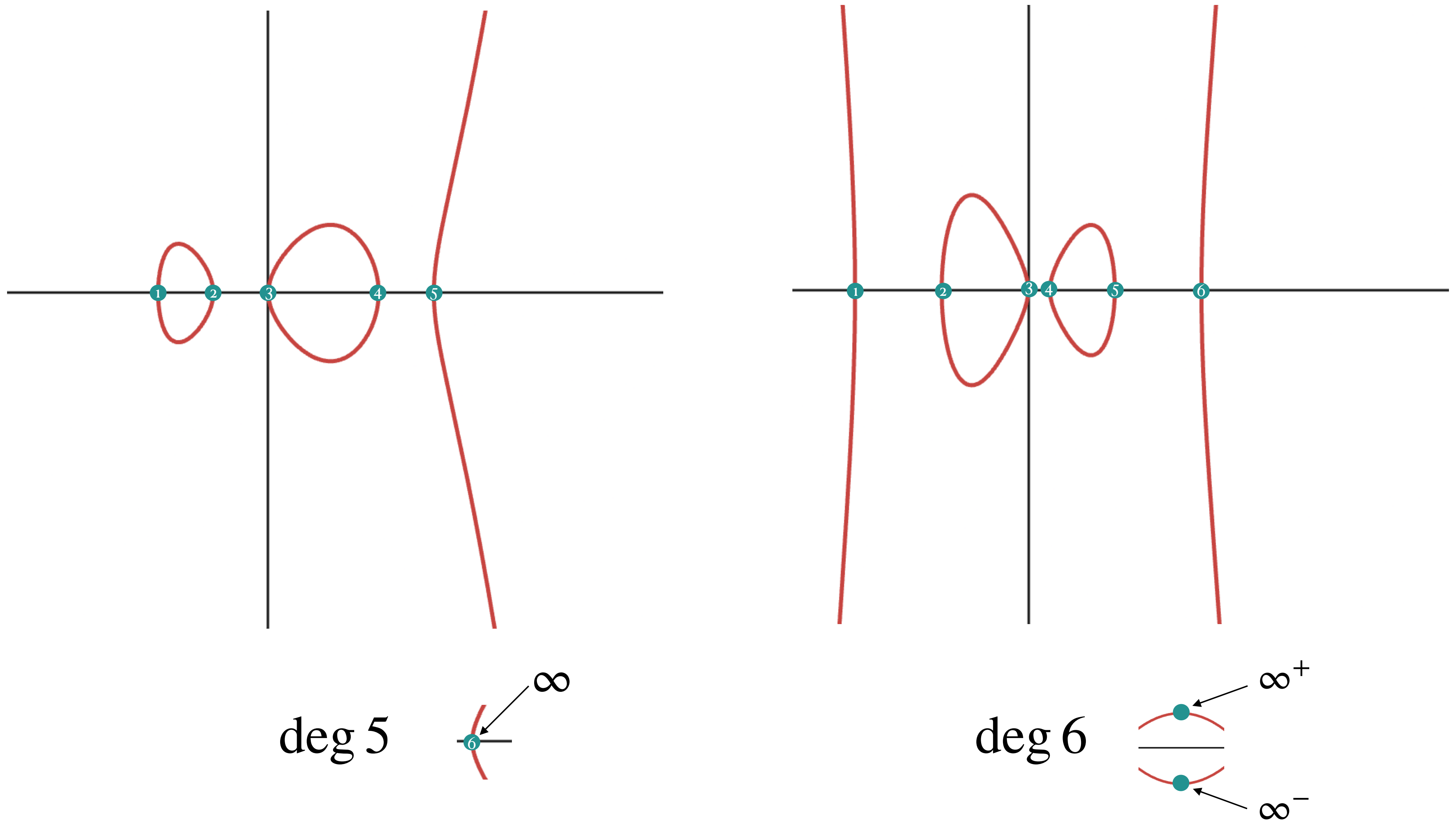
Elliptic Curves

$$E\colon y^2 = x^3 + Ax + B$$



Genus-2 Hyperelliptic Curve

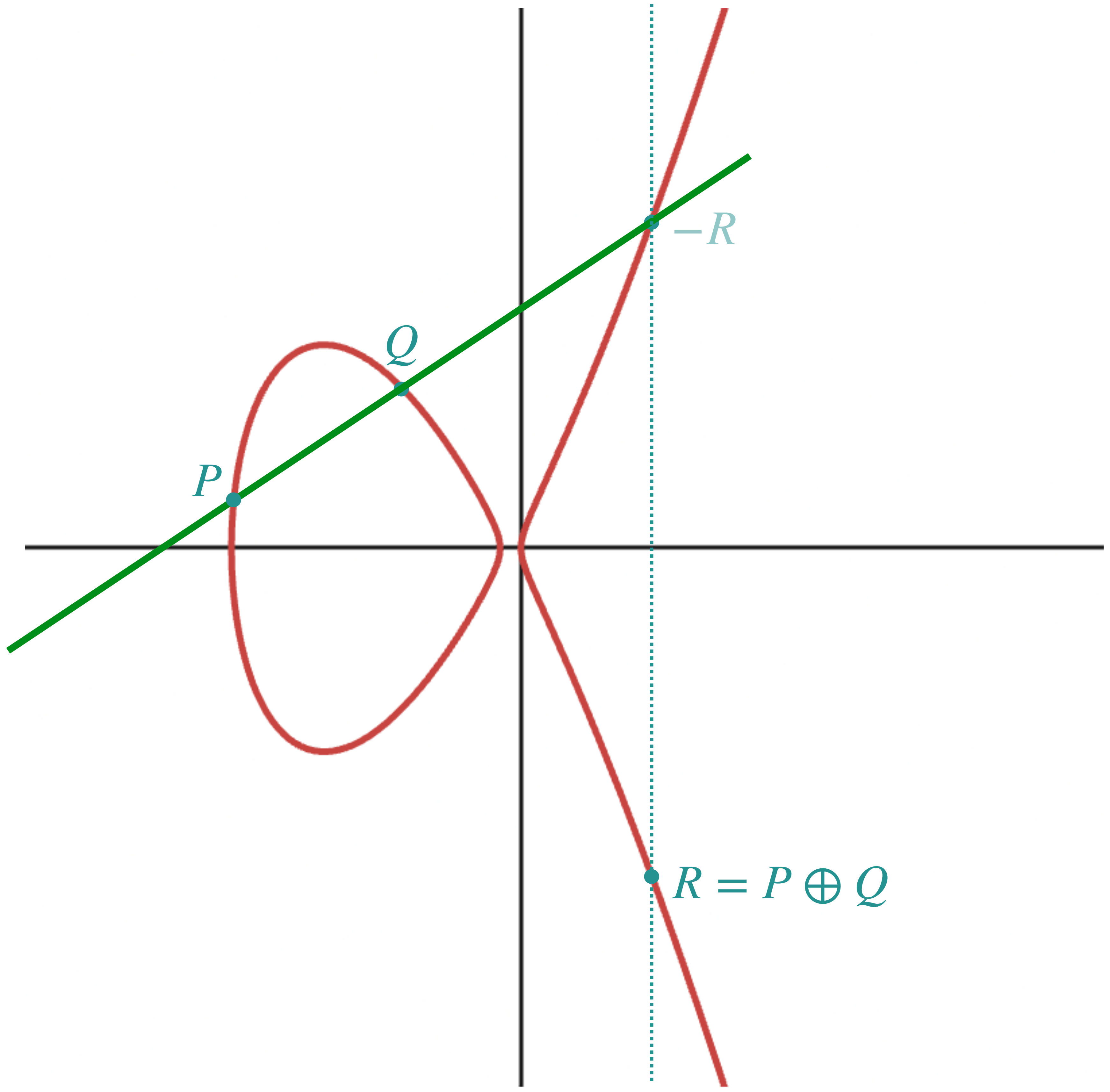
$$C\colon y^2 = f(x), \quad \deg(f) = 5 \text{ or } 6$$



Moving to higher dimension

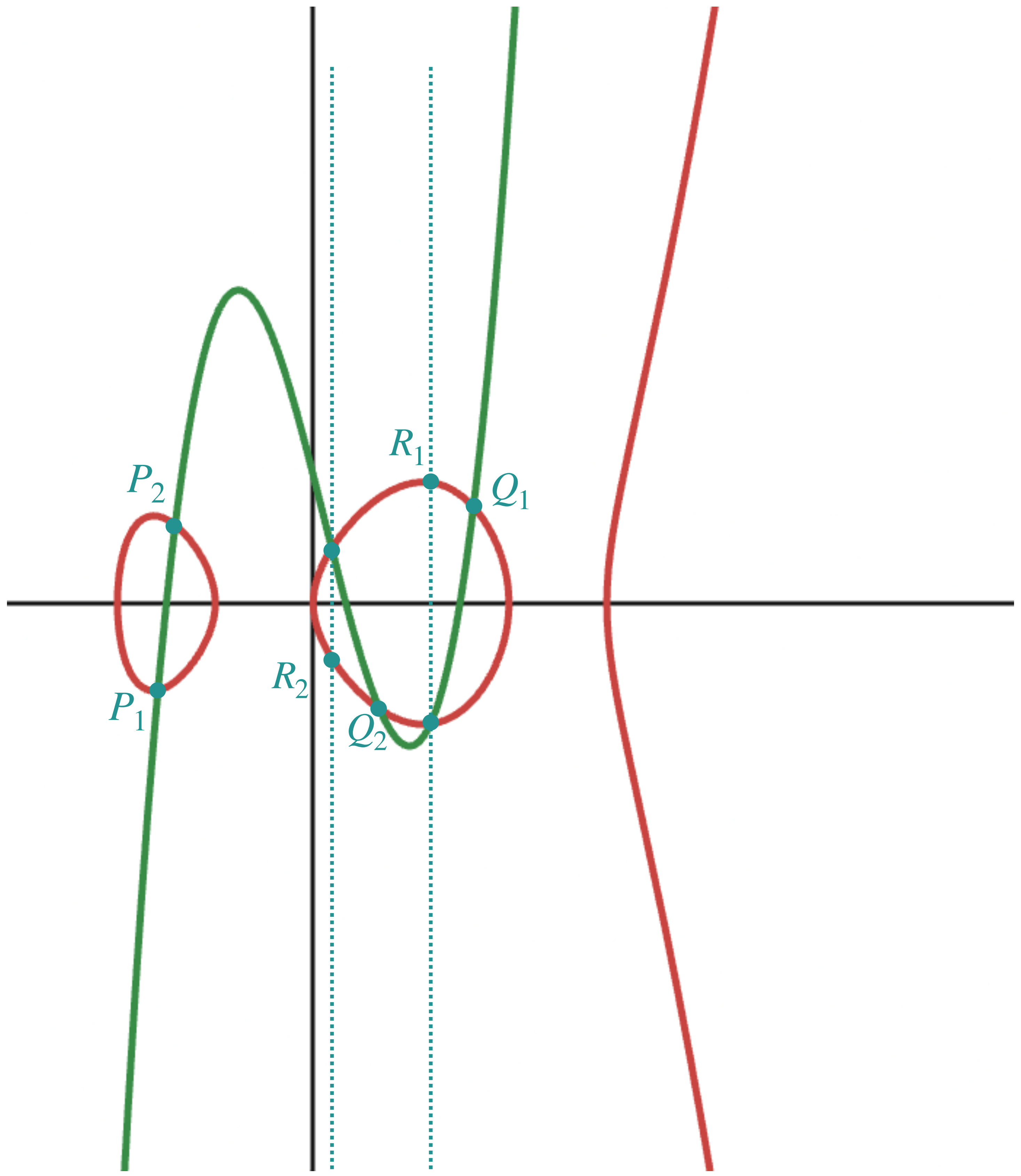
Elliptic Curves

$E\colon y^2 = x^3 + Ax + B$



Genus-2 Hyperelliptic Curve

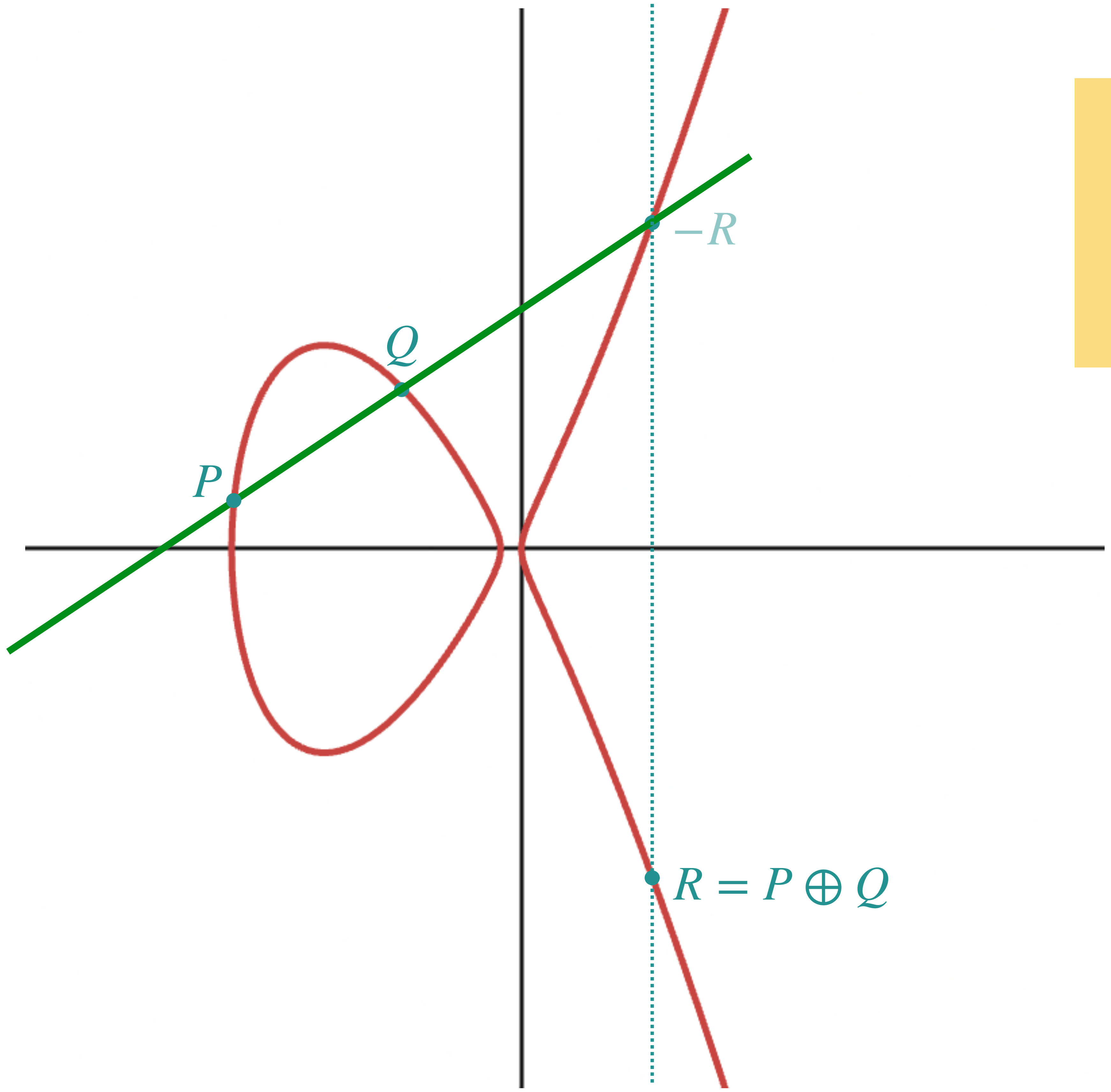
$C\colon y^2 = f(x), \quad \deg(f) = 5 \text{ or } 6$



Moving to higher dimension

Elliptic Curves

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

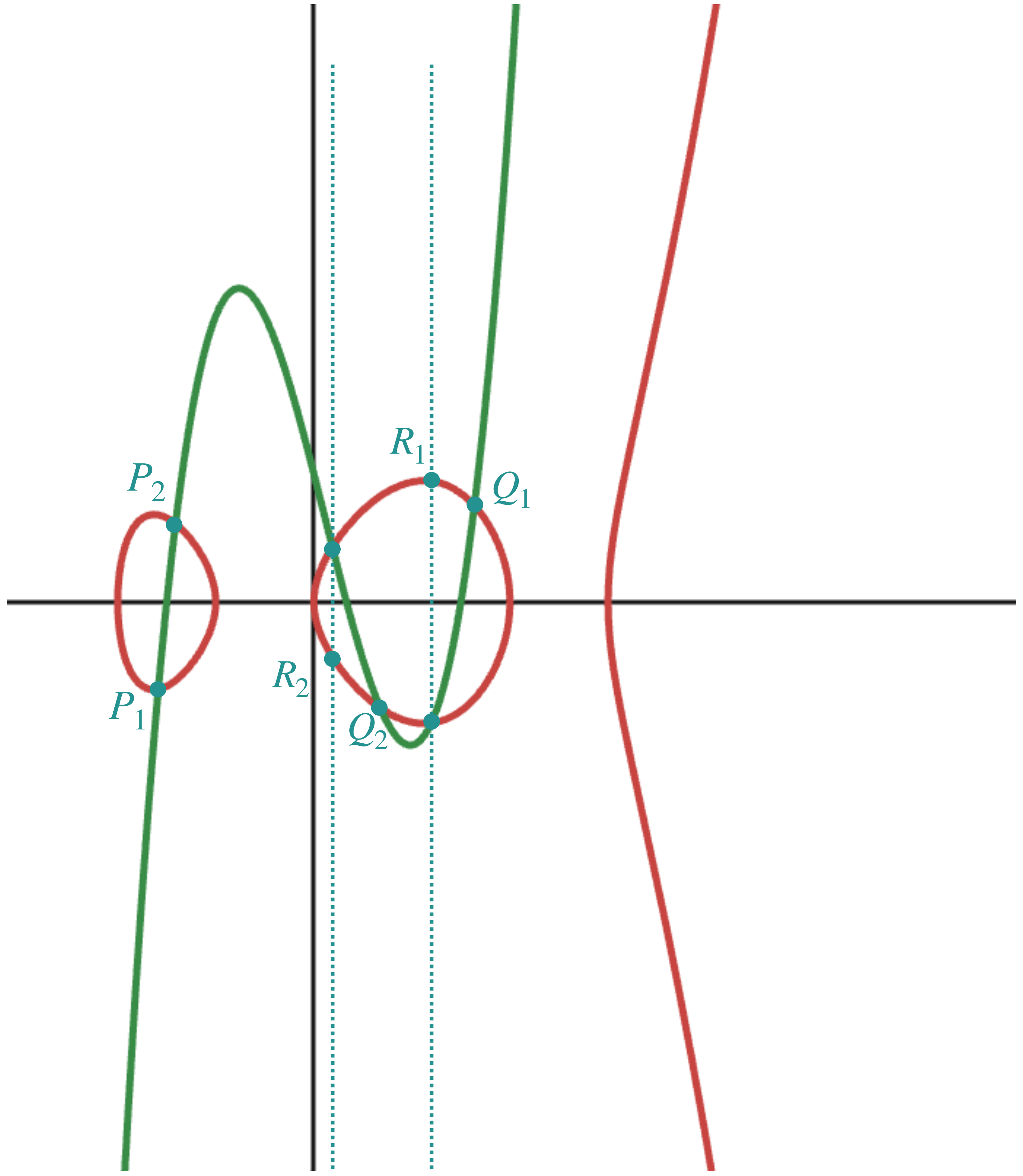


The Jacobian is the group attached to C.

In genus 1, $J_C \cong C$.

Genus-2 Hyperelliptic Curve

$C: y^2 = f(x), \quad \deg(f) = 5 \text{ or } 6$



Moving to higher dimension

We now consider **superspecial** (principally polarised) **abelian surfaces** (A, λ).

There are two types:

- Products of elliptic curves $E_1 \times E_2$
- Jacobians J_C of genus 2 hyperelliptic curves C

Definition

An isogeny $\Phi : A_1 \to A_2$ between PPAS is a surjective group homomorphism with finite kernel.

We say it is a (polarised) ℓ-isogeny if:

- $\ker(\Phi) \cong (\mathbb{Z}/\ell\mathbb{Z})^2$
- The isogeny respects the polarisations

Moving to higher dimension

We now consider **superspecial** **(principally polarised)** **abelian surfaces** (A, λ).

There are two types:

- Products of elliptic curves $E_1 \times E_2$
- Jacobians J_C of genus 2 hyperelliptic curves C

Definition

An isogeny $\Phi : A_1 \to A_2$ between PPAS is a surjective group homomorphism with finite kernel.

We say it is a (polarised) ℓ-isogeny if:

- $\ker(\Phi) = \langle R, S \rangle$ where $R, S \in A[\ell]$ and $e_\ell(R, S) = 1$

Types of isogenies

Gluing

$$E_1 \times E_2 \to J_C$$

Generic

$$J_1 \to J_2$$

Splitting

$$J_C \to E_1 \times E_2$$

Types of isogenies

Gluing

$E_1 \times E_2 \to J_C$

Generic

$J_1 \to J_2$

Splitting

$J_C \to E_1 \times E_2$

gluing 2-isogeny

$$\underbrace{E_1 \times E_2 \to J_1}_{\text{gluing 2-isogeny}} \to \ldots \to J_k \to E_3 \times E_4$$

Types of isogenies

Gluing

$$E_1 \times E_2 \to J_C$$

Generic

$$J_1 \to J_2$$

Splitting

$$J_C \to E_1 \times E_2$$

generic 2-isogenies

$$E_1 \times E_2 \to J_1 \to \ldots \to J_k \to E_3 \times E_4$$

Types of isogenies

Gluing

$$E_1 \times E_2 \to J_C$$

Generic

$$J_1 \to J_2$$

Splitting

$$J_C \to E_1 \times E_2$$

splitting 2-isogeny

$$E_1 \times E_2 \to J_1 \to \ldots \to J_k \to E_3 \times E_4$$

Types of isogenies

Gluing

$E_1 \times E_2 \to J_C$

Generic

$J_1 \to J_2$

Splitting

$J_C \to E_1 \times E_2$

$$E_1 \times E_2 \to J_1 \to \ldots \to J_k \to E_3 \times E_4$$

Let's focus on this for now

III. Kummer Surfaces

Motivation: the Kummer line

Models of Elliptic Curves

Short Weierstrass

$$E\colon y^2 = x^3 + Ax + B$$

Motivation: the Kummer line

Models of Elliptic Curves

Short Weierstrass

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

If we have rational
2-torsion on E $\cong$

Montgomery Form

$$E: y^2 = x(x - \alpha)(x - 1/\alpha)$$

Motivation: the Kummer line

Models of Elliptic Curves

Short Weierstrass

$$E\colon y^2 = x^3 + Ax + B$$

If we have rational 2-torsion on E $\cong$

Montgomery Form

$$E\colon y^2 = x(x-\alpha)(x-1/\alpha)$$

$$\xrightarrow{\pi}$$

$$\pi : P \mapsto \begin{cases} (1:0) \text{ if } P = \mathcal{O} \\ (x:1) \text{ if } P = (x,y) \end{cases}$$

Montgomery Kummer Line

$$E/\langle \pm 1 \rangle \hookrightarrow \mathbb{P}^1$$

Many Kummer Line models

The Montgomery Model $(X : Z)$

$$\mathcal{O} = (1 : 0),\ T_1 = (0 : 1),$$
$$T_2 = (a : b),\ T_3 = (b : a)\,.$$

The Theta Model $(\theta_0 : \theta_1)$

$$\mathcal{O} = (a : b),\ T_1 = (-a : b),$$
$$T_2 = (b : a),\ T_3 = (-b : a)\,.$$

Many Kummer Line models

$$(X : Z) \mapsto (a(X - Z) : b(X + Z))$$

The Montgomery Model $(X : Z)$

$$\mathcal{O} = (1 : 0),\ T_1 = (0 : 1),$$
$$T_2 = (a : b),\ T_3 = (b : a)\,.$$

The Theta Model $(\theta_0 : \theta_1)$

$$\mathcal{O} = (a : b),\ T_1 = (-a : b),$$
$$T_2 = (b : a),\ T_3 = (-b : a)\,.$$

Many Kummer Line models

$$(X : Z) \mapsto (a(X - Z) : b(X + Z))$$

The Montgomery Model $(X : Z)$

$\mathcal{O} = (1 : 0),\ T_1 = (0 : 1),$

$T_2 = (a : b),\ T_3 = (b : a)\,.$

The Theta Model $(\theta_0 : \theta_1)$

$\mathcal{O} = (a : b),\ T_1 = (-a : b),$

$T_2 = (b : a),\ T_3 = (-b : a)\,.$

$$(\theta_0 : \theta_1) \mapsto (a\theta_1 + b\theta_0 : a\theta_1 - b\theta_0)$$

Kummer Surfaces

Models of Hyperelliptic Curves

Genus-2 Hyperelliptic Curve

$$C\colon y^2 = f(x), \quad \deg(f) = 5 \text{ or } 6$$

Kummer Surfaces

Models of Hyperelliptic Curves

Genus-2 Hyperelliptic Curve

$$C: y^2 = f(x), \quad \deg(f) = 5 \text{ or } 6$$

If we have rational Weierstrass points on C $\cong$

Rosenhain Form

$$C_{\lambda,\mu,\nu}: y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu)$$

Kummer Surfaces

Models of Hyperelliptic Curves

Genus-2 Hyperelliptic Curve

$C\colon y^2 = f(x), \quad \deg(f) = 5 \text{ or } 6$

If we have rational Weierstrass points on C $\cong$

Rosenhain Form

$C_{\lambda,\mu,\nu}\colon y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu)$

$\xrightarrow{\pi : J_C \to J_C/\langle \pm 1 \rangle \hookrightarrow \mathbb{P}^3}$

Cassels—Flynn Kummer surfaces

Cassels and Flynn. *Prolegomena to a Middlebrow Arithmetic of Curves of Genus 2* (1996).

Kummer Surfaces

Models of Hyperelliptic Curves

Genus-2 Hyperelliptic Curve

$C: y^2 = f(x), \quad \deg(f) = 5 \text{ or } 6$

$\xrightarrow{\pi : J_C \to J_C/\langle \pm 1 \rangle \hookrightarrow \mathbb{P}^3}$

Cassels—Flynn Kummer surfaces

If we have rational Weierstrass points on C $\cong$

Rosenhain Form

$C_{\lambda,\mu,\nu}: y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu)$

$\xrightarrow{\pi : J_C \to J_C/\langle \pm 1 \rangle \hookrightarrow \mathbb{P}^3}$

Kummer surfaces arising from theta functions of level 2

Gaudry. *Fast genus 2 arithmetic based on Theta functions* (2007).

The theta model

The Kummer surface can be embedded into $\mathbb{P}^3$ with theta coordinates. The system of theta coordinates (of level 2) is fully determined by the *theta null point* $\mathcal{O} = (a : b : c : d)$.

The theta model

The Kummer surface can be embedded into $\mathbb{P}^3$ with theta coordinates. The system of theta coordinates (of level 2) is fully determined by the *theta null point* $\mathcal{O} = (a : b : c : d)$.

Two-torsion points

$$\begin{array}{cccc}
(a : b : c : d) & (a : -b : c : -d) & (a : b : -c : -d) & (a - : b : -c : d) \\
(b : a : d : c) & (b : -a : d : -c) & (b : a : -d : -c) & (b : -a : -d : c) \\
(c : d : a : b) & (c : -d : a : -b) & (c : d : -a : -b) & (c : -d : -a : b) \\
(d : c : b : a) & (d : -c : b : -a) & (d : c : -b : -a) & (d : -c : -b : a)
\end{array}$$

Symmetry in the 2-torsion gives us fast doubling and 2-isogenies!

The theta model

The Kummer surface can be embedded into $\mathbb{P}^3$ with theta coordinates. The system of theta coordinates (of level 2) is fully determined by the *theta null point* $\mathcal{O} = (a : b : c : d)$.

Two-torsion points

$$
\begin{array}{llll}
(a : b : c : d) & (a : -b : c : -d) & (a : b : -c : -d) & (a - : b : -c : d) \\
(b : a : d : c) & (b : -a : d : -c) & (b : a : -d : -c) & (b : -a : -d : c) \\
(c : d : a : b) & (c : -d : a : -b) & (c : d : -a : -b) & (c : -d : -a : b) \\
(d : c : b : a) & (d : -c : b : -a) & (d : c : -b : -a) & (d : -c : -b : a)
\end{array}
$$

There are different systems of theta coordinates and appropriate change of coordinates formulas; we fix a specific choice.

The theta model: arithmetic

The group law is destroyed by quotienting by $\langle \pm 1 \rangle$ but we still have a *pseudo-group law*.

Fast arithmetic : built from simple building blocks

$$\mathrm{H} : (X_1 : X_2 : X_3 : X_4) \mapsto (X_1 + X_2 + X_3 + X_4 : X_1 + X_2 - X_3 - X_4 : X_1 - X_2 + X_3 - X_4 : X_1 - X_2 - X_3 + X_4)$$

The theta model: arithmetic

The group law is destroyed by quotienting by $\langle \pm 1 \rangle$ but we still have a *pseudo-group law*.

Fast arithmetic : built from simple building blocks

$$\mathrm{H} : (X_1 : X_2 : X_3 : X_4) \mapsto (X_1 + X_2 + X_3 + X_4 : X_1 + X_2 - X_3 - X_4 : X_1 - X_2 + X_3 - X_4 : X_1 - X_2 - X_3 + X_4)$$

$$\mathrm{S} : (X_1 : X_2 : X_3 : X_4) \mapsto (X_1^2 : X_2^2 : X_3^2 : X_4^2)$$

The theta model: arithmetic

The group law is destroyed by quotienting by $\langle \pm 1 \rangle$ but we still have a *pseudo-group law*.

Fast arithmetic : built from simple building blocks

$$\mathrm{H} : (X_1 : X_2 : X_3 : X_4) \mapsto (X_1 + X_2 + X_3 + X_4 : X_1 + X_2 - X_3 - X_4 : X_1 - X_2 + X_3 - X_4 : X_1 - X_2 - X_3 + X_4)$$

$$\mathrm{S} : (X_1 : X_2 : X_3 : X_4) \mapsto (X_1^2 : X_2^2 : X_3^2 : X_4^2)$$

$$\mathrm{C}_U : (X_1 : X_2 : X_3 : X_4) \mapsto (X_1 \cdot U_1 : X_2 \cdot U_2 : X_3 \cdot U_3 : X_4 \cdot U_4)$$

IV. Computing Generic Isogenies

Dartois, Maino, Pope and Robert. *An Algorithmic Approach to (2,2)-isogenies in the Theta Model* (2024).

The theta model: isogenies

A 2-isogeny $J \to J'$ induces an algebraic map between Kummer surfaces.

$$
\begin{array}{ccc}
J & \longrightarrow & J' \\
\downarrow & & \downarrow \\
J/\langle \pm 1 \rangle & \xrightarrow{\Phi} & J'/\langle \pm 1 \rangle
\end{array}
$$

The theta model: isogenies

A 2-isogeny $J \to J'$ induces an algebraic map between Kummer surfaces.

$$\begin{tikzcd} J \arrow[r] \arrow[d] & J' \arrow[d] \\ J/\langle \pm 1 \rangle \arrow[r, "\Phi"] & J'/\langle \pm 1 \rangle \end{tikzcd}$$

“We lose nothing by going down to the Kummers” - Cassels and Flynn

The theta model: isogenies

A 2-isogeny $J \to J'$ induces an algebraic map between Kummer surfaces.

$$
\begin{tikzcd}
J \arrow[r] \arrow[d] & J' \arrow[d] \\
J/\langle \pm 1 \rangle \arrow[r, "\Phi"] & J'/\langle \pm 1 \rangle
\end{tikzcd}
$$

“Computing (generic) isogenies between abelian surfaces with theta coordinates of level 2”

The theta model: isogenies

Let A, B be abelian surfaces in (a specific choice of) level-2 theta coordinates.

Key Fact:

Let $\Phi : A \to B$ be a 2-isogeny where $\mathcal{O}_A = (a : b : c : d)$. Then

$$\mathcal{O}_B = \mathsf{H} \circ \mathsf{S}^{-1} \circ \mathsf{H} \circ \mathsf{S}(\mathcal{O}_A)$$

Computing generic isogenies

Algorithm 8.34 $\mathsf{GenericCodomain}(0_A)$

Input: The theta constants $0_A := (a : b : c : d)$ of A.

Output: Dual isogeneous theta null point $(\alpha : \beta : \gamma : \delta)$, the inverse of the dual isogeneous theta null point $(\alpha^{-1} : \beta^{-1} : \gamma^{-1} : \delta^{-1})$, and the theta null point 0_B on B. // Case $\alpha \cdot \beta \cdot \gamma \cdot \delta \neq 0$.

$(\alpha^2, \beta^2, \gamma^2, \delta^2) \leftarrow \mathcal{H} \circ \mathcal{S}(a, b, c, d)$

$\alpha \leftarrow \alpha^2$

$\beta \leftarrow \alpha^2 \cdot \beta^2$

$\gamma \leftarrow \alpha^2 \cdot \gamma^2$

$\delta \leftarrow \alpha^2 \cdot \delta^2$

$\beta \leftarrow \mathsf{SquareRoot}(\beta)$

$\gamma \leftarrow \mathsf{SquareRoot}(\gamma)$

$\delta \leftarrow \mathsf{SquareRoot}(\delta)$

$\alpha^{-1} \leftarrow \gamma^2 \cdot \delta^2$

$\beta^{-1} \leftarrow \alpha^{-1} \cdot \beta$

$\alpha^{-1} \leftarrow \alpha^{-1} \cdot \beta^2$

$\gamma^{-1} \leftarrow \delta^2 \cdot \beta^2 \cdot \gamma$

$\delta^{-1} \leftarrow \gamma^2 \cdot \beta^2 \cdot \delta$

$(a_2, b_2, c_2, d_2) \leftarrow \mathcal{H}(\alpha, \beta, \gamma, \delta)$

$0_B \leftarrow (a_2, b_2, c_2, d_2)$

return $(\alpha, \beta, \gamma, \delta), (\alpha^{-1}, \beta^{-1}, \gamma^{-1}, \delta^{-1}), 0_B$ // Total cost: $4\mathbf{S} + 10\mathbf{M} + 24\mathbf{a} + 3\mathbf{Sqrt}$

Computing generic isogenies

Algorithm 8.34 GenericCodomain(0_A)

Input: The theta constants $0_A := (a : b : c : d)$ of A.

Output: Dual isogeneous theta null point $(\alpha : \beta : \gamma : \delta)$, the inverse of the dual isogeneous theta null point $(\alpha^{-1} : \beta^{-1} : \gamma^{-1} : \delta^{-1})$, and the theta null point 0_B on B. // Case $\alpha \cdot \beta \cdot \gamma \cdot \delta \neq 0$.

Otherwise, A would correspond to a product of elliptic curves

$(\alpha^2, \beta^2, \gamma^2, \delta^2) \leftarrow \mathcal{H} \circ \mathcal{S}(a, b, c, d)$
$\alpha \leftarrow \alpha^2$
$\beta \leftarrow \alpha^2 \cdot \beta^2$
$\gamma \leftarrow \alpha^2 \cdot \gamma^2$
$\delta \leftarrow \alpha^2 \cdot \delta^2$
$\beta \leftarrow \mathsf{SquareRoot}(\beta)$
$\gamma \leftarrow \mathsf{SquareRoot}(\gamma)$
$\delta \leftarrow \mathsf{SquareRoot}(\delta)$
$\alpha^{-1} \leftarrow \gamma^2 \cdot \delta^2$
$\beta^{-1} \leftarrow \alpha^{-1} \cdot \beta$
$\alpha^{-1} \leftarrow \alpha^{-1} \cdot \beta^2$
$\gamma^{-1} \leftarrow \delta^2 \cdot \beta^2 \cdot \gamma$
$\delta^{-1} \leftarrow \gamma^2 \cdot \beta^2 \cdot \delta$
$(a_2, b_2, c_2, d_2) \leftarrow \mathcal{H}(\alpha, \beta, \gamma, \delta)$
$0_B \leftarrow (a_2, b_2, c_2, d_2)$
return $(\alpha, \beta, \gamma, \delta), (\alpha^{-1}, \beta^{-1}, \gamma^{-1}, \delta^{-1}), 0_B$ // Total cost: $4\mathbf{S} + 10\mathbf{M} + 24\mathbf{a} + 3\mathbf{Sqrt}$

Computing generic isogenies

Algorithm 8.34 $\mathsf{GenericCodomain}(0_A)$

Input: The theta constants $0_A := (a : b : c : d)$ of A.

Output: Dual isogeneous theta null point $(\alpha : \beta : \gamma : \delta)$, the inverse of the dual isogeneous theta null point $(\alpha^{-1} : \beta^{-1} : \gamma^{-1} : \delta^{-1})$, and the theta null point 0_B on B. // Case $\alpha \cdot \beta \cdot \gamma \cdot \delta \neq 0$.

$(\alpha^2, \beta^2, \gamma^2, \delta^2) \leftarrow \mathcal{H} \circ \mathcal{S}(a, b, c, d)$

$\alpha \leftarrow \alpha^2$

$\beta \leftarrow \alpha^2 \cdot \beta^2$

$\gamma \leftarrow \alpha^2 \cdot \gamma^2$

$\delta \leftarrow \alpha^2 \cdot \delta^2$

$\beta \leftarrow \mathsf{SquareRoot}(\beta)$

$\gamma \leftarrow \mathsf{SquareRoot}(\gamma)$

$\delta \leftarrow \mathsf{SquareRoot}(\delta)$

Computing S^{-1}

$\alpha^{-1} \leftarrow \gamma^2 \cdot \delta^2$

$\beta^{-1} \leftarrow \alpha^{-1} \cdot \beta$

$\alpha^{-1} \leftarrow \alpha^{-1} \cdot \beta^2$

$\gamma^{-1} \leftarrow \delta^2 \cdot \beta^2 \cdot \gamma$

$\delta^{-1} \leftarrow \gamma^2 \cdot \beta^2 \cdot \delta$

$(a_2, b_2, c_2, d_2) \leftarrow \mathcal{H}(\alpha, \beta, \gamma, \delta)$

$0_B \leftarrow (a_2, b_2, c_2, d_2)$

return $(\alpha, \beta, \gamma, \delta), (\alpha^{-1}, \beta^{-1}, \gamma^{-1}, \delta^{-1}), 0_B$ // Total cost: $4\mathbf{S} + 10\mathbf{M} + 24\mathbf{a} + 3\mathbf{Sqrt}$

Computing generic isogenies

Algorithm 8.34 $\mathsf{GenericCodomain}(0_A)$

Input: The theta constants $0_A := (a : b : c : d)$ of A.

Output: Dual isogeneous theta null point $(\alpha : \beta : \gamma : \delta)$, the inverse of the dual isogeneous theta null point $(\alpha^{-1} : \beta^{-1} : \gamma^{-1} : \delta^{-1})$, and the theta null point 0_B on B. // Case $\alpha \cdot \beta \cdot \gamma \cdot \delta \neq 0$.

$(\alpha^2, \beta^2, \gamma^2, \delta^2) \leftarrow \mathcal{H} \circ \mathcal{S}(a, b, c, d)$

$\alpha \leftarrow \alpha^2$

$\beta \leftarrow \alpha^2 \cdot \beta^2$

$\gamma \leftarrow \alpha^2 \cdot \gamma^2$

$\delta \leftarrow \alpha^2 \cdot \delta^2$

$\beta \leftarrow \mathsf{SquareRoot}(\beta)$

$\gamma \leftarrow \mathsf{SquareRoot}(\gamma)$

$\delta \leftarrow \mathsf{SquareRoot}(\delta)$

$\alpha^{-1} \leftarrow \gamma^2 \cdot \delta^2$

$\beta^{-1} \leftarrow \alpha^{-1} \cdot \beta$

$\alpha^{-1} \leftarrow \alpha^{-1} \cdot \beta^2$

$\gamma^{-1} \leftarrow \delta^2 \cdot \beta^2 \cdot \gamma$

$\delta^{-1} \leftarrow \gamma^2 \cdot \beta^2 \cdot \delta$

$(a_2, b_2, c_2, d_2) \leftarrow \mathcal{H}(\alpha, \beta, \gamma, \delta)$

$0_B \leftarrow (a_2, b_2, c_2, d_2)$

return $(\alpha, \beta, \gamma, \delta), (\alpha^{-1}, \beta^{-1}, \gamma^{-1}, \delta^{-1}), 0_B$ // Total cost: $4\mathbf{S} + 10\mathbf{M} + 24\mathbf{a} + 3\mathbf{Sqrt}$

What we've done so far

Codomain computation

$$E_1 \times E_2 \to A_1 \to \dots \to A_k \to E_3 \times E_4$$

What we've done so far

Codomain computation

$$E_1 \times E_2 \to A_1 \to \dots \to A_k \to E_3 \times E_4$$

But square roots are *expensive*…

Removing square roots

In analogy with dimension one, torsion points allow us to avoid square roots!

Removing square roots

In analogy with dimension one, torsion points allow us to avoid square roots!

Fun Fact

Let R', S' be such that $\ker(\Phi) = \langle [4]R', [4]S' \rangle$. Then:

$$\mathsf{H} \circ \mathsf{S}(R') = (x\alpha : x\beta : y\gamma : y\delta)$$
$$\mathsf{H} \circ \mathsf{S}(S') = (z\alpha : w\beta : z\gamma : w\delta)$$

for some $x, y, z, w \in \mathbb{F}_{p^2}$.

Removing square roots

In analogy with dimension one, torsion points allow us to avoid square roots!

Fun Fact

Let R', S' be such that $\ker(\Phi) = \langle [4]R', [4]S' \rangle$. Then:

$$\mathsf{H} \circ \mathsf{S}(R') = (x\alpha : x\beta : y\gamma : y\delta)$$
$$\mathsf{H} \circ \mathsf{S}(S') = (z\alpha : w\beta : z\gamma : w\delta)$$

for some $x, y, z, w \in \mathbb{F}_{p^2}$.

This tells us we can extract $(\alpha : \beta : \gamma : \delta)$ using a few squaring and multiplications.

Removing square roots

In analogy with dimension one, torsion points allow us to avoid square roots!

Fun Fact

Let R', S' be such that $\ker(\Phi) = \langle [4]R', [4]S' \rangle$. Then:

$$\mathrm{H} \circ \mathrm{S}(R') = (x\alpha : x\beta : y\gamma : y\delta)$$
$$\mathrm{H} \circ \mathrm{S}(S') = (z\alpha : w\beta : z\gamma : w\delta)$$

for some $x, y, z, w \in \mathbb{F}_{p^2}$.

This tells us we can extract $(\alpha : \beta : \gamma : \delta)$ using a few squaring and multiplications.

We can do something similar with 4-torsion, but we need 2 square roots.

Evaluating points

We compute images of points in an analogous way.

Let $\Phi : A \to B$ be a 2-isogeny where $\mathcal{O}_A = (a : b : c : d)$. Then the image of a point P is given by

$$\mathsf{H} \circ \mathsf{C} \circ \mathsf{H} \circ \mathsf{S}(P)$$

where $\mathsf{C} : (x : y : z : w) \mapsto (x \cdot \alpha^{-1} : y \cdot \beta^{-1} : z \cdot \gamma^{-1} : w \cdot \delta^{-1})$.

We have already computed
$(\alpha^{-1} : \beta^{-1} : \gamma^{-1} : \delta^{-1})$

What we've done so far

Codomain computation and evaluation

$$E_1 \times E_2 \to A_1 \to \ldots \to A_k \to E_3 \times E_4$$

Without square roots!

What we've done so far

Codomain computation and evaluation

$$E_1 \times E_2 \to A_1 \to \ldots \to A_k \to E_3 \times E_4$$

Without square roots!

But we need gluings and splittings!

V. Gluing and splitting isogenies

Splitting isogenies

1

Generic Isogeny
$\varphi : A \to B$
where $B \cong E_1 \times E_2$

Splitting isogenies

1

Generic Isogeny
$\varphi : A \to B$
where $B \cong E_1 \times E_2$

2

Compute isomorphism whose action on $\mathcal{O}_B$ gives back the theta null point $\mathcal{O}_{E_1 \times E_2}$ associated with the *product theta structure.*

Splitting isogenies

1

Generic Isogeny
$\varphi : A \to B$
where $B \cong E_1 \times E_2$

2

Compute isomorphism whose action on $\mathcal{O}_B$ gives back the theta null point $\mathcal{O}_{E_1 \times E_2}$ associated with the *product theta structure.*

Product Theta Structure

Let $B = E \times E'$ be a product of Montgomery elliptic curves each with their system of theta coordinates

$$(\theta_0 : \theta_1) \quad \text{and} \quad (\theta'_0 : \theta'_1)$$

The *product theta coordinates* on B are defined as

$$(x : y : z : w) = (\theta_0\theta'_0 : \theta_1\theta'_0 : \theta_0\theta'_1 : \theta_1\theta'_1)$$

Splitting isogenies

1

Generic Isogeny
$\varphi : A \to B$
where $B \cong E_1 \times E_2$

2

Compute isomorphism whose action on $\mathcal{O}_B$ gives back the theta null point $\mathcal{O}_{E_1 \times E_2}$ associated with the *product theta structure.*

3

Recover Montgomery coefficients of E_1, E_2 from $\mathcal{O}_{E_1 \times E_2}$.

Recall...

The Montgomery Model $(X : Z)$

$$\mathcal{O} = (1 : 0),\ T_1 = (0 : 1),$$
$$T_2 = (a : b),\ T_3 = (b : a)\,.$$

The Theta Model $(\theta_0 : \theta_1)$

$$\mathcal{O} = (a : b),\ T_1 = (-a : b),$$
$$T_2 = (b : a),\ T_3 = (-b : a)\,.$$

$$(\theta_0 : \theta_1) \mapsto (a\theta_1 + b\theta_0 : a\theta_1 - b\theta_0)$$

Splitting isogenies

1

Generic Isogeny
$\varphi : A \to B$
where $B \cong E_1 \times E_2$

2

Compute isomorphism whose action on $\mathcal{O}_B$ gives back the theta null point $\mathcal{O}_{E_1 \times E_2}$ associated with the *product theta structure.*

3

Recover Montgomery coefficients of E_1, E_2 from $\mathcal{O}_{E_1 \times E_2}$.

Procedure is similar for points…

Gluing isogenies

1

Apply change of basis matrix to move points $(P_1, P_2) \in E_1 \times E_2$ from Montgomery coordinates to (non-product) theta coordinates (corresp. to our choice of theta structure)

Gluing isogenies

1

Apply change of basis matrix to move points $(P_1, P_2) \in E_1 \times E_2$ from Montgomery coordinates to (non-product) theta coordinates (corresp. to our choice of theta structure)

Product to Theta Structure

Let $E \times E'$ be a product of Montgomery elliptic curves each with product theta coordinates

$$(x : y : z : w) = (\theta_0\theta_0' : \theta_1\theta_0' : \theta_0\theta_1' : \theta_1\theta_1')$$

Then apply a change of basis matrix.

Gluing isogenies

1

Apply change of basis matrix to move points $(P_1, P_2) \in E_1 \times E_2$ from Montgomery coordinates to (non-product) theta coordinates (corresp. to our choice of theta structure)

2

Compute the codomain of the gluing isogeny

$$E_1 \times E_2 \to A$$

(taking into account we always have $\delta = 0$).

Gluing isogenies

1

Apply change of basis matrix to move points $(P_1, P_2) \in E_1 \times E_2$ from Montgomery coordinates to (non-product) theta coordinates (corresp. to our choice of theta structure)

2

Compute the codomain of the gluing isogeny

$$E_1 \times E_2 \to A$$

(taking into account we always have $\delta = 0$).

Procedure is similar for evaluating points (P_1, P_2)…
Special (faster) evaluation for points of the form $(P_1,0)$ or $(0,P_2)$

VI. Open Questions

Open Questions

(thanks also to Luciano and Pierrick!)

?

How can we best treat cases when a splitting (and therefore then a gluing) happens in the middle of the chain? How can we treat cases where gluing does not happen on the first step?

?

?

?

Open Questions

(thanks also to Luciano and Pierrick!)

How can we best treat cases when a splitting (and therefore then a gluing) happens in the middle of the chain? How can we treat cases where gluing does not happen on the first step?

Can we develop new side channel attacks against these algorithms? For example, by generalising the patient-zero attack to two dimensions. Can this be used as a side-channel attack against SQIsign?

Open Questions

(thanks also to Luciano and Pierrick!)

?

How can we best treat cases when a splitting (and therefore then a gluing) happens in the middle of the chain? How can we treat cases where gluing does not happen on the first step?

?

Can we develop new side channel attacks against these algorithms? For example, by generalising the patient-zero attack to two dimensions. Can this be used as a side-channel attack against SQIsign?

?

Two-dimensional isogenies of odd degree? SqrtVélu2D?

?

Open Questions

(thanks also to Luciano and Pierrick!)

? How can we best treat cases when a splitting (and therefore then a gluing) happens in the middle of the chain? How can we treat cases where gluing does not happen on the first step?

? Can we develop new side channel attacks against these algorithms? For example, by generalising the patient-zero attack to two dimensions. Can this be used as a side-channel attack against SQIsign?

? Two-dimensional isogenies of odd degree? SqrtVélu2D?

? Improve gluing! (coming up next…)

Resources (besides the spec)

An Algorithmic Approach to (2,2)-isogenies in the Theta Model

Dartois, Maino, Pope and Robert

Some notes on algorithms for abelian varieties

Robert

Isogenies on Kummer Surfaces

Joint with Flynn