# Representing abelian varieties

Stefano Marseglia

Utrecht University

Welcome Home 2019 - Universitá di Torino

# Abelian Varieties

- An abelian variety $A$ over a field $k$ is a projective geometrically connected group variety over $k$.

## Abelian Varieties

- An abelian variety $A$ over a field $k$ is a projective geometrically connected group variety over $k$.
  We have morphisms $\oplus : A \times A \to A$, $\ominus : A \to A$ and a $k$-rational point $e \in A(k)$ such that $(A, \oplus, \ominus, e)$ is a group object in the category of projective geom. connected varieties over $k$.

# Abelian Varieties

- An abelian variety $A$ over a field $k$ is a projective geometrically connected group variety over $k$.
  We have morphisms $\oplus : A \times A \to A$, $\ominus : A \to A$ and a $k$-rational point $e \in A(k)$ such that $(A, \oplus, \ominus, e)$ is a group object in the category of projective geom. connected varieties over $k$.
- In practice, we have diagrams $\rightsquigarrow$ "natural" group structure on $A(\overline{k})$.

# Abelian Varieties

- An abelian variety $A$ over a field $k$ is a projective geometrically connected group variety over $k$.
  We have morphisms $\oplus : A \times A \to A$, $\ominus : A \to A$ and a $k$-rational point $e \in A(k)$ such that $(A, \oplus, \ominus, e)$ is a group object in the category of projective geom. connected varieties over $k$.
- In practice, we have diagrams $\rightsquigarrow$ "natural" group structure on $A(\overline{k})$.
- eg. ($\ominus$ is the "inverse" morphism)

$$\begin{tikzcd}
A \times_k A \arrow[rr, "(\ominus,\mathrm{id})"] & & A \times_k A \arrow[d, "\oplus"] \\
A \arrow[u, "\Delta"] \arrow[r] & \mathrm{Spec}(k) \arrow[r, "e"] & A
\end{tikzcd}$$

# Abelian Varieties

- An abelian variety $A$ over a field $k$ is a projective geometrically connected group variety over $k$.
  We have morphisms $\oplus : A \times A \to A$, $\ominus : A \to A$ and a $k$-rational point $e \in A(k)$ such that $(A, \oplus, \ominus, e)$ is a group object in the category of projective geom. connected varieties over $k$.
- In practice, we have diagrams $\rightsquigarrow$ "natural" group structure on $A(\overline{k})$.
- eg. ($\ominus$ is the "inverse" morphism)

$$\begin{tikzcd}
A \times_k A \arrow[rr, "(\ominus,\mathrm{id})"] & & A \times_k A \arrow[d, "\oplus"] \\
A \arrow[u, "\Delta"] \arrow[r] & \mathrm{Spec}(k) \arrow[r, "e"] & A
\end{tikzcd}
\qquad
\begin{tikzcd}
A \times_k A \arrow[rr, "(\mathrm{id},\ominus)"] & & A \times_k A \arrow[d, "\oplus"] \\
A \arrow[u, "\Delta"] \arrow[r] & \mathrm{Spec}(k) \arrow[r, "e"] & A
\end{tikzcd}$$

## Example : $\dim A = 1$ elliptic curves

- AVs of dimension 1 are called Elliptic Curves.

## Example : $\dim A = 1$ elliptic curves

- AVs of dimension 1 are called Elliptic Curves.
- They admit a plane model: if $\operatorname{char} k \neq 2, 3$

$$Y^2 Z = X^3 + AXZ^2 + BZ^3 \quad A, B \in k \text{ and } e = [0:1:0]$$

## Example : $\dim A = 1$ elliptic curves

- AVs of dimension 1 are called Elliptic Curves.
- They admit a plane model: if $\operatorname{char} k \neq 2, 3$

$$Y^2 Z = X^3 + AXZ^2 + BZ^3 \quad A, B \in k \text{ and } e = [0:1:0]$$

- The groups law is explicit:
  if $P = (x_P, y_P)$ then $-P = (x_P, -y_P)$ and

## Example : $\dim A = 1$ elliptic curves

- AVs of dimension 1 are called Elliptic Curves.
- They admit a plane model: if $\operatorname{char} k \neq 2, 3$

$$Y^2 Z = X^3 + AXZ^2 + BZ^3 \quad A, B \in k \text{ and } e = [0:1:0]$$

- The groups law is explicit:
  if $P = (x_P, y_P)$ then $-P = (x_P, -y_P)$ and
  if $Q = (x_Q, y_Q) \neq -P$ then $P + Q = (x_R, y_R)$ where

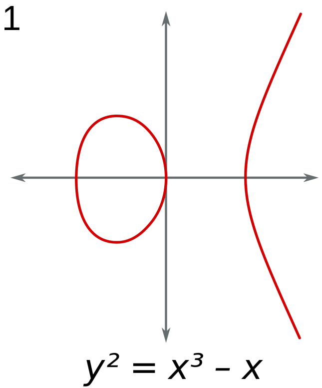
$$x_R = \lambda^2 - x_P - x_Q, \quad y_R = y_P + \lambda(x_R - x_P),$$

  where

$$\lambda = \begin{cases} \frac{3x_P^2 + B}{2A} & \text{if } P = Q \\ \frac{y_P - y_Q}{x_P - x_Q} & \text{if } P \neq Q \end{cases}.$$

## Example : $\dim A = 1$ elliptic curves

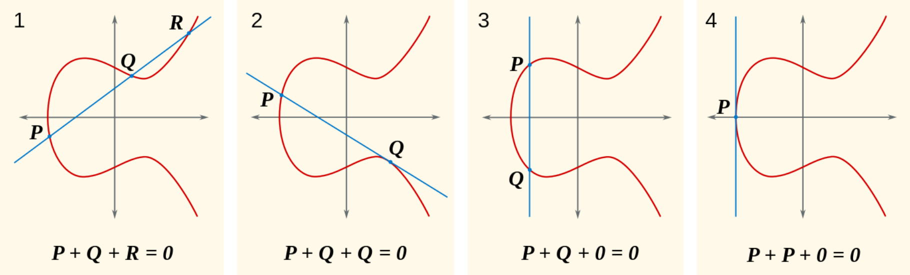
In pictures, over $k = \mathbb{R}$:




source: `https://commons.wikimedia.org/wiki/File:ECClines-3.svg`

# Example : $\dim A = 1$ elliptic curves

Group law in pictures, over $k = \mathbb{R}$:




source: `https://commons.wikimedia.org/wiki/File:ECClines.svg`

# In higher dimension

- If $\dim A > 1$ we have equations ($A$ is projective!)...

## In higher dimension

- If $\dim A > 1$ we have equations ($A$ is projective!)...
- ...but in general they are really complicated.

# In higher dimension

- If $\dim A > 1$ we have equations ($A$ is projective!)...
- ...but in general they are really complicated.
- In 1990 Flynn: equations for a "general" abelian surface (in $\mathrm{char}(k) \neq 2,3,5$).

## In higher dimension

- If $\dim A > 1$ we have equations ($A$ is projective!)...
- ...but in general they are really complicated.
- In 1990 Flynn: equations for a "general" abelian surface (in $\mathrm{char}(k) \neq 2,3,5$).
- they look like this:

```
# This file is available by anon. ftp from ftp.liv.ac.uk as
#     ~ftp/pub/genus2/jacobian.variety/defining.equations
#
# The following 72 quadratic forms eqn(1),..,eqn(72) give a set
# of defining equations for the jacobian variety of the
# curve of genus 2:
# Y**2 = f6*X**6 + f5*X**5 + f4*X**4 + f3*X**3 + f2*X**2 + f1*X + f0
# using the embedding into projective 15-space give by
# the following functions on the divisor
#        {(x,y),(u,v)} = (x,y) + (u,v) - infty+ - infty-.
# Note that a0,a3,a4,a5,a10,a11,a12,a13,a14,a15 are even
# and a1,a2,a6,a7,a8,a9 are odd.
# The defining equations have been organised so that
# eqn(1)..eqn(21) are purely even*even terms
# eqn(22)..eqn(42) have odd*odd and even*even terms
# and eqn(43)..eqn(72) have purely odd*even terms.
#
# a15 := (x-u)**2; a14 := 1; a13 := x + u; a12 := x*u;
# a11 := x*u*(x+u); a10 := (x*u)**2; a9 := (y-v)/(x-u);
# a8 := (u*y-x*v)/(x-u); a7 := (u**2*y-x**2*v)/(x-u);
# a6 := (u**3*y-x**3*v)/(x-u); a5 := (f0xu-2*y*v)/((x-u)**2);
# a4 := (f1xu-(x+u)*y*v)/((x-u)**2); a3 := (x*u)*a5;
# ## where f0xu, f1xu are
# f0xu := 2*f0+f1*(x+u)+2*f2*(x*u)+f3*(x+u)*(x*u)
#              +2*f4*(x*u)**2+f5*(x+u)*(x*u)**2+2*f6*(x*u)**3;
# f1xu := f0*(x+u)+2*f1*(x*u)+f2*(x+u)*(x*u)+2*f3*(x*u)**2
#              +f4*(x+u)*(x*u)**2+2*f5*(x*u)**3+f6*(x+u)*(x*u)**3;
# a2:=(gxu*y-gux*v)/((x-u)**3); a1:=(hxu*y-hux*v)/((x-u)**3);
# where gxu, gux, hxu, hux are
# gxu := f0*4+f1*(x+3*u)+f2*(2*x*u+2*u**2)+f3*(3*x*u**2+u**3)
#          +f4*(4*x*u**3)+f5*x*(x*u**3+3*u**4)+f6*2*x*(x*u**4+u**5);
# gux := f0*4+f1*(u+3*x)+f2*(2*u*x+2*x**2)+f3*(3*u*x**2+x**3)
#          +f4*(4*u*x**3)+f5*u*(u*x**3+3*x**4)+f6*2*u*(u*x**4+x**5);
# hxu := f0*2*(x+u)+f1*u*(3*x+u)+f2*4*x*u**2+f3*x*u**2*(x+3*u)
#         +f4*2*x*u**3*(x+u)+f5*x*u**4*(3*x+u)+f6*4*x**2*u**5;
# hux := f0*2*(u+x)+f1*x*(3*u+x)+f2*4*u*x**2+f3*u*x**2*(u+3*x)
#          +f4*2*u*x**3*(u+x)+f5*u*x**4*(3*u+x)+f6*4*u**2*x**5;
# a0 :=a5**2;

eqn(1) := -a0*a11+f1*a14*a3+f3*a10*a5+f5*a3*a10+2*a4*a3;
eqn(2) := -a0*a10+a3**2;
eqn(3) := -a0*a12+a3*a5;
eqn(4) := -f0*f2*a14**2-f0*a14*a5-8*f0*f6*a12**2-f3*f5*a12*a10-
f1*f6*
a13*a10-f2*f5*a13*a10-f1*f5*a13*a11-3*f5*f0*a13*a12-f1*f3*a14*a12-
f3*f0*a14*a13
-f0*f6*a14*a10-f2*f4*a14*a10+a4**2-a0*a12-6*f0*f6*a12*a15-
f2*f6*a10*a15-f1*f6*
a11*a15-f5*f0*a13*a15-f1*f4*a14*a11-f2*a12*a5-f4*f0*a13**2-
f0*f6*a15**2-f4*f6*
a10**2-f6*a10*a3-f4*a10*a5-
f3*f6*a10*a11-4*f2*f6*a10*a12-2*f1*f6*a11*a12;
eqn(5) := -a0*a13+f1*a14*a5+f3*a14*a3+f5*a10*a5+2*a5*a4;
eqn(6) := -a0*a14+a5**2;
eqn(7) :=
-4*f0*f2*a14**2-4*f0*a14*a5+a0*a15-36*f0*f6*a12**2-4*f3*f5*
a12*a10-4*f1*f6*a13*a10-4*f2*f5*a13*a10-12*f5*f0*a13*a12-2*f1*f3*a14
*a12-4*f3*
f0*a14*a13-4*f2*f4*a14*a10-4*f5*a10*a4-
f5**2*a10**2-24*f0*f6*a12*a15-4*f2*f6*
a10*a15-4*f1*f6*a11*a15-4*f5*f0*a13*a15-4*f1*f4*a14*a11+f3**2*a14*a1
0-2*f3*a11*
a5-16*f1*f5*a12**2-2*f1*a13*a5-4*f2*a12*a5-4*f0*f6*a15**2-4*f4*f6*a1
0**2-4*f6*
a10*a3-4*f4*a10*a5-4*f3*f6*a10*a11-16*f2*f6*a10*a12-8*f1*f6*a11*a12+
f1**2*a14**
2-16*f0*f4*a14*a12-4*f1*f5*a12*a15-4*f0*f4*a14*a15;
eqn(8) := -f1*a14**2-f3*a14*a12+2*f4*a13*a12-
f5*a12**2-2*a4*a14-2*f4*
a14*a11+a5*a13;
eqn(9) := -f1*a14*a12-f3*a14*a10-f5*a12*a10-2*a4*a12+a5*a11;
eqn(10) := 2*f4*a14*a10+2*f5*a12*a11-4*a5*a12-2*f4*a12**2-a5*a15+f1*
a14*a13+f3*a14*a11-f5*a13*a10+2*a4*a13;
eqn(11) := -f5*a10**2-f3*a10*a12+2*f2*a11*a12-
f1*a12**2-2*a4*a10-2*f2*
a13*a10+a3*a11;
eqn(12) := f2*a12**2+f1*a13*a12-a5*a10-f1*a14*a11-f2*a10*a14+a3*a12;
eqn(13) := f5*a13*a10+f4*a14*a10-a5*a12-f4*a12**2-f5*a12*a11+a3*a14;
eqn(14) := -f1*a14*a12-f3*a14*a10-f5*a12*a10-2*a4*a12+a3*a13;
eqn(15) := 4*f1*a13*a12-2*a5*a10-3*f1*a14*a11-
a3*a15-2*a3*a12+f5*a10*
a11+f3*a10*a13+2*a4*a11;
eqn(16) := -a14*a15-4*a12*a14+a13**2;
eqn(17) := -a10*a14+a12**2;
eqn(18) := -a10*a15-4*a10*a12+a11**2;
eqn(19) := -a11*a13+2*a10*a14+a12*a15+2*a12**2;
eqn(20) := -a12*a13+a11*a14;
eqn(21) := -a11*a12+a10*a13;
eqn(22) := -a10**2*f2*f5**2-a11**2*f0*f5**2+a1**2-
a0*a3+8*f0*f6*a4*a11
-a10**2*f3**2*f6-f4*a3**2-
f0*a5**2+4*a10**2*f1*f5*f6+4*a10**2*f2*f4*f6-a10*a3*
f3*f5+4*a10*a3*f2*f6+8*f1*f6*a10*a4+f1*f5*a10*a5+4*a11**2*f0*f4*f6-
a10*a11*f1*
f5**2+4*a10*a11*f0*f5*f6+4*a10*a11*f1*f4*f6+4*f0*f5*a12*a4+2*a12*a10
*f0*f5**2+6
*a12*a10*f1*f3*f6+8*f0*f3*f6*a12*a11+4*a14*a10*f0*f2*f6+2*a14*a10*f0
*f3*f5+3*
a14*a10*f1**2*f6+4*f0*f1*f6*a14*a11+2*f0*f1*f5*a14*a12;
eqn(23) := a1*a2-
a0*a4+3*a13*a10*f0*f5**2+a13*a10*f1*f3*f6+2*a10**2*f2
*f5*f6+f3*f6*a10*a3+4*f2*f6*a10*a4+a10*a5*f2*f5+5*a10*a5*f1*f6+4*f1*
f6*a12*a3+
20*f0*f6*a12*a4+10*f0*f5*f6*a10*a12+2*a12**2*f1*f3*f5+28*a12**2*f0*f
3*f6+4*a12
**2*f1*f2*f6+3*f1*f5*a12*a4+2*a12*a10*f1*f4*f6+2*a12*a10*f2*f3*f6+a1
2*a10*f1*f5
**2-4*f0*f5**2*a12*a11-4*f0*f6*f4*a12*a11+2*f0*f4*a13*a5+8*a10*a13*f
```

```
0*f4*f6+8*
a13*a12*f0*f2*f6+3*a13*a12*f0*f3*f5-
a13*a12*f1**2*f6+9*a14*a3*f0*f5+a14*a3*f1*
f4+f0*f3*a14*a5-2*f1*f6*f2*a14*a10-8*f0*f6*f3*a14*a10-2*f0*f5*f3*a14
*a11-4*f0*
f6*f2*a14*a11+10*f1*f6*f0*a14*a12+2*a14*a12*f0*f2*f5+a14*a12*f1**2*f
5+2*f1*f6*
a3*a15+4*f0*f6*a4*a15+2*f0*f5*a5*a15+2*f0*f5*f6*a10*a15+4*f0*f3*f6*a
12*a15+2*f1
*f6*f0*a14*a15;
eqn(24) := -a14**2*f0*f3**2-a14**2*f1**2*f4+a2**2-a0*a5-f6*a3**2-
f2*a5
**2+8*a13*a4*f0*f6+4*f0*f5*a13*a5+4*a13*a10*f0*f5*f6-4*a13*a10*f1*f4
*f6+4*f0*f4
*a14*a5+4*a10*a14*f0*f4*f6-
a10*a14*f0*f5**2+4*a10*a14*f1*f3*f6+8*f1*f6*a12*a4+4
*f1*f5*f6*a12*a10+4*f1*f6*f4*a12*a11-2*f1*f6*a13*a3+4*a14*a13*f0*f1*
f6+4*a14*
a13*f0*f2*f5-a14*a13*f1**2*f5+4*a14**2*f0*f2*f4+f1*f5*a14*a3-
a14*a5*f1*f3+8*a14
*a11*f0*f3*f6+16*a14*a12*f0*f2*f6+2*a14*a12*f0*f3*f5+4*a14*f0*f2*f6*
a15-a14*f1
**2*f6*a15;
eqn(25) := -a0*a14-f2*a14*a5-f3*a14*a4-2*f4*a14*a3-3*f5*a4*a12-
f6*a3*
a15-5*f6*a3*a12-f1*f3*a14**2-f1*f4*a14*a13-
f1*f5*a14*a15-5*f1*f5*a12*a14-f1*f6*
a13*a15-3*f1*f6*a13*a12-2*f2*f4*a14*a12-2*f2*f5*a13*a12-2*f2*f6*a13*
a11-3*f3*f5
*a14*a10-2*f3*f6*a12*a11-2*f4*f6*a12*a10-f5**2*a12*a10+a2*a9;
eqn(26) := -a0*a13+f1*a14*a5+f3*a5*a12-
f5*a10*a5-2*f6*a11*a3+2*f0*f3*
a14**2+4*f0*f4*a14*a13+4*f5*f0*a14*a15+14*f5*f0*a14*a12+4*f0*f6*a13*
a15+8*f0*f6
*a13*a12+4*f0*f6*a14*a11+2*f1*f4*a14*a12+2*f1*f5*a13*a12+2*f1*f6*a12
*a15+8*f1*
f6*a12**2+2*a2*a8;
eqn(27) := 2*f2*a3*a14-
a0*a15+40*f0*f6*a12**2+6*f3*f5*a12*a10+4*f2*f5*
a13*a10+8*f5*f0*a13*a12+3*f1*f3*a14*a12+2*f3*f0*a14*a13+8*f0*f6*a14*
a10+4*f2*f4
*a14*a10-2*a0*a12+4*f5*a10*a4+f5**2*a10**2+4*f3*a12*a4+28*f0*f6*a12*
a15+4*f1*f6
*a11*a15+4*f5*f0*a13*a15+4*f1*f4*a14*a11+f3**2*a14*a10+4*f2*f6*a11**
2+16*f1*f5*
a12**2+f1*a13*a5+2*a2*a7+4*f0*f6*a15**2+4*f4*f6*a10**2+2*f6*a10*a3+4
*f4*a10*a5+
4*f3*f6*a10*a11+14*f1*f6*a11*a12+16*f0*f4*a14*a12+4*f1*f5*a12*a15+4*
f0*f4*a14*
a15+f1*f5*a14*a10+6*a14*a11*f0*f5;
eqn(28) := -a0*a11-4*f0*a13*a5-f1*a15*a5-5*f1*a12*a5-2*f2*a11*a5-f3*
a10*a5+f5*a3*a10-4*f0*f2*a14*a13-2*f3*f0*a15*a14-10*f0*f3*a14*a12-4*
f0*f4*a14*
a11-2*a15*a12*f0*f5-6*f0*f5*a12**2+f1**2*a14*a13-
f1*f3*a14*a11-2*f1*f4*a12**2-
f1*f5*a13*a10+2*a2*a6;
eqn(29) := -a0*a10-f4*a3*a10-f3*a4*a10-2*f2*a10*a5-3*f1*a12*a4-
f0*a15*
a5-5*f0*a12*a5-f5*f3*a10**2-f5*f2*a10*a11-
f1*f5*a15*a10-5*f1*f5*a10*a12-a15*a11
*f0*f5-3*f5*f0*a11*a12-2*f4*f2*a10*a12-2*f4*f1*a11*a12-2*f0*f4*a13*a
11-3*a14*
a10*f1*f3-2*f3*f0*a13*a12-2*f0*f2*a14*a12-f1**2*a14*a12+a6*a1;
eqn(30) := -a0*a11+f5*a3*a10+f3*a3*a12-
f1*a14*a3-2*f0*a13*a5+2*f6*f3*
a10**2+4*f2*f6*a10*a11+4*a10*f1*f6*a15+14*a10*a12*f1*f6+4*f0*f6*a15*
a11+8*f0*f6
*a12*a11+4*f0*f6*a13*a10+2*f2*f5*a12*a10+2*f1*f5*a12*a11+2*a15*a12*f
0*f5+8*f0*
f5*a12**2+2*a1*a7;
eqn(31) := 4*f0*f2*a14**2+2*f0*a14*a5+f5*a3*a11-
a0*a15+40*f0*f6*a12**2
+3*f3*f5*a12*a10+6*f1*f6*a13*a10+14*f5*f0*a13*a12+6*f1*f3*a14*a12+4*
f3*f0*a14*
a13+8*f0*f6*a14*a10-2*a0*a12+4*f3*a12*a4+28*f0*f6*a12*a15+4*f2*f6*a1
0*a15+4*f1*
f6*a11*a15+4*f5*f0*a13*a15+4*f1*f4*a14*a11+f3**2*a14*a10+2*a1*a8+16*
f1*f5*a12**
2+4*f2*a12*a5+4*f4*f0*a13**2+4*f0*f6*a15**2+2*f4*a10*a5+2*f3*f6*a10*
a11+16*f2*
f6*a10*a12+8*f1*f6*a11*a12+f1**2*a14**2+4*f1*f5*a12*a15+f1*f5*a14*a1
0+4*f2*f4*
a12**2+4*f1*a14*a4+2*a12*a11*f3*f4+4*f2*f5*a12*a11-2*a10*a13*f3*f4-2
*a13*a12*f2
*f3+2*a14*a11*f2*f3;
eqn(32) := -a0*a13-4*f6*a11*a3-f5*a3*a15-5*f5*a3*a12-2*f4*a13*a3-f3*
a14*a3+f1*a14*a5-4*f6*f4*a10*a11-2*f6*f3*a10*a15-10*f6*f3*a10*a12-4*
f2*f6*a13*
a10-2*f1*f6*a12*a15-6*f1*f6*a12**2+f5**2*a10*a11-
f3*f5*a13*a10-2*f5*f2*a12**2-
f1*f5*a14*a11+2*a1*a9;
eqn(33) := -a5*a14-f2*a14**2-f3*a14*a13-f4*a13**2-3*f5*a13*a12-
f5*a13*
a15-f6*a14*a10-6*f6*a12*a15-8*f6*a12**2-f6*a15**2+a9**2;
eqn(34) := a9*a8-a4*a14-f3*a14*a12-f4*a14*a11-
f5*a12*a15-4*f5*a12**2-
f6*a11*a15-2*f6*a11*a12-f6*a13*a10;
eqn(35) := 2*a9*a7-
a5*a15-2*a5*a12+f1*a14*a13+2*f2*a14*a12+f3*a14*a11-
f5*a13*a10-2*f6*a15*a10-6*f6*a10*a12;
eqn(36) := a6*a9-a4*a15-
a4*a12+f2*a14*a11+2*f3*a14*a10+f4*a12*a11+f1*
a14*a12+f5*a12*a10;
eqn(37) := a8**2-a5*a12-f0*a14**2-f4*a12**2-f5*a12*a11-
f6*a15*a10-4*f6
*a10*a12;
eqn(38) := a7*a8-a4*a12-f0*a14*a13-f1*a14*a12-f5*a12*a10-f6*a10*a11;
eqn(39) := 2*a6*a8-
```

```
a3*a15-2*a3*a12+f5*a10*a11+2*f4*a12*a10+f3*a10*a13-
f1*a14*a11-2*f0*a15*a14-6*f0*a14*a12;
eqn(40) := a7**2-a5*a10-f0*a15*a14-4*f0*a14*a12-f1*a14*a11-
f2*a10*a14-
f6*a10**2;
eqn(41) := a6*a7-a4*a10-f3*a10*a12-f2*a13*a10-
f1*a12*a15-4*f1*a12**2-
f0*a15*a13-2*f0*a13*a12-f0*a11*a14;
eqn(42) := -a3*a10-f4*a10**2-f3*a11*a10-f2*a11**2-3*f1*a11*a12-
f1*a15*
a11-f0*a14*a10-6*f0*a15*a12-8*f0*a12**2-f0*a15**2+a6**2;
eqn(43) := -f1*a9*a3+a14*a8*f1**2+4*f1*a8*a4+2*f3*a3*a7+a3*a1-f3*a2*
a10+4*f0*a8*a5-
a6*a0+2*f2*a8*a3-2*f0*f5*a10*a9+3*f1*f5*a10*a8+2*f1*f6*a11*a6+12
*f0*f5*a12*a7+12*f0*f6*a12*a6+4*a12*a8*f0*f4+2*a12*a8*f1*f3+2*a12*a7
*f1*f4+2*
a12*a6*f1*f5+4*a14*a8*f0*f2+4*f0*f3*a14*a7+4*f0*f4*a14*a6+4*f0*f5*a7
*a15+4*f0*
f6*a6*a15;
eqn(44) := -a7*a0+2*f0*a9*a5+f1*a8*a5+a2*a3;
eqn(45) := f5*a10*a2+a2*a4+a14*a8*f2**2-a0*a8+f6*a6*a3-f2*a9*a4-
f2*f4*
a11*a9+a11*a8*f1*f6-
a11*a8*f2*f5+4*f2*f4*a12*a8+4*a12*a7*f2*f5-4*a12*a7*f1*f6+3
*f2*f6*a12*a6+f0*f5*a12*a9-f0*f3*a14*a9-a14*a8*f1*f3+a14*a7*f2*f3-
a14*a7*f1*f4-
f1*f5*a14*a6+f2*f4*a8*a15+a7*f2*f5*a15-a7*f1*f6*a15+f2*f6*a6*a15;
eqn(46) := -
f3**2*a14*a7-2*a10*a7*f5**2+4*f6*a7*a3+a2*a5+4*f1*f6*a11*
a9-
a0*a9+f3*a9*a4+3*f5*a3*a8+2*f4*a7*a5+4*f0*f5*a13*a9-2*f5*f6*a10*a6+4
*a10*a7*
f4*f6+4*f3*f6*a10*a8+4*f2*f6*a10*a9+12*f0*f6*a12*a9-
f3*f6*a12*a6+4*a12*a8*f2*f5
+4*a12*a9*f1*f5+4*a12*a7*f2*f6+2*a12*a7*f3*f5-a12*a8*f3*f4-
f3*f5*a13*a6+2*f1*f5
*a14*a7-a14*a6*f3*f4-a14*a8*f2*f3+2*a14*a6*f1*f6+4*f0*f6*a9*a15-
f3*f6*a6*a15;
eqn(47) := -a0*a8+2*f6*a6*a3+f5*a3*a7+a5*a1;
eqn(48) := f0*a9*a5+a1*a4-f4*f2*a13*a6+a10*a7*f4**2+f1*a14*a1-
f4*a6*a4
+f6*f1*a12*a6-a7*a0+a13*a7*f0*f5-a13*a7*f1*f4-f6*f3*a10*a6-
a10*a7*f3*f5+a10*a8*
f3*f4-a10*a8*f2*f5-
f5*f1*a10*a9+4*f2*f4*a12*a7+4*a12*a8*f1*f4-4*f0*f5*a12*a8+3*
f4*f0*a12*a9+f2*f4*a7*a15+a8*f1*f4*a15-f0*f5*a8*a15+f4*f0*a9*a15;
eqn(49) := -
a9*a5+f3*a14*a8+2*f4*a14*a7+2*f5*a14*a6+2*f6*a13*a6+f5*a8*
a12+a2*a14;
eqn(50) := -2*a5*a8-f1*a14*a9-2*f2*a14*a8-
f3*a14*a7+f5*a7*a12+2*f6*a12
*a6+a2*a13;
eqn(51) := -a5*a7+2*f0*a14*a9+f1*a14*a8+a2*a12;
eqn(52) := -a3*a7+2*f0*a12*a9+f1*a12*a8+a2*a10;
eqn(53) := -2*a8*a3-f1*a12*a9-2*f2*a12*a8-
f3*a12*a7+f5*a10*a7+2*f6*a10
*a6+a2*a11;
eqn(54) := -2*a5*a7-f1*a13*a9-2*f2*a14*a7-2*f2*a12*a9-
f3*a14*a6-3*f3*
a8*a12-4*f4*a12*a7-3*f5*a12*a6-
f5*a10*a8-2*f6*a11*a6+a2*a15+2*a2*a12;
eqn(55) := -
a3*a6+f3*a10*a7+2*f2*a10*a8+2*f1*a10*a9+2*f0*a11*a9+f1*a12
*a7+a1*a10;
eqn(56) := -2*a3*a7-f5*a10*a6-2*f4*a10*a7-
f3*a10*a8+f1*a12*a8+2*f0*a12
*a9+a1*a11;
eqn(57) := -a8*a3+2*f6*a10*a6+f5*a10*a7+a1*a12;
eqn(58) := -a5*a8+2*f6*a12*a6+f5*a7*a12+a1*a14;
eqn(59) := -2*a5*a7-f5*a12*a6-2*f4*a12*a7-
f3*a8*a12+f1*a14*a8+2*f0*a14
*a9+a1*a13;
eqn(60) := -2*a8*a3-f5*a11*a6-2*f4*a8*a10-2*f4*a12*a6-
f3*a10*a9-3*f3*
a12*a7-4*f2*a12*a8-3*f1*a12*a9-
f1*a14*a7-2*f0*a13*a9+a1*a15+2*a1*a12;
eqn(61) := -
a9*a4+f2*a14*a8+f3*a14*a7+f4*a14*a6+f4*a12*a8+f5*a13*a6+f6
*a6*a15+3*f6*a12*a6+a5*a8;
eqn(62) := -a4*a8-f0*a14*a9-
f1*a14*a8+f4*a12*a7+f5*a12*a6+f6*a11*a6+a5
*a7;
eqn(63) := -a4*a7-f0*a13*a9-f1*a14*a7-f1*a12*a9-f2*a12*a8-
f3*a12*a7+f6
*a10*a6+a6*a5;
eqn(64) := -
a4*a6+f4*a10*a7+f3*a10*a8+f2*a10*a9+f2*a7*a12+f1*a11*a9+f0
*a15*a9+3*f0*a12*a9+a3*a7;
eqn(65) := -a4*a7-f6*a10*a6-
f5*a10*a7+f2*a12*a8+f1*a12*a9+f0*a13*a9+a8
*a3;
eqn(66) := -a4*a8-f6*a11*a6-f5*a10*a8-f5*a12*a6-f4*a12*a7-
f3*a8*a12+f0
*a14*a9+a9*a3;
eqn(67) := -4*a7*a12-a7*a15+a8*a11+a6*a13;
eqn(68) := -4*a8*a12-a8*a15+a7*a13+a9*a11;
eqn(69) := -a8*a13+a7*a14+a9*a12;
eqn(70) := -a7*a13+a6*a14+a8*a12;
eqn(71) := -a8*a11+a9*a10+a7*a12;
eqn(72) := -a7*a11+a8*a10+a6*a12;
```

# In higher dimension

- 72 equations in 16 variables.....
- (reference: Flynn-Cassels “Prolegomena to a Middlebrow Arithmetic of Curves of Genus 2”)

# Other representations

- Jacobian of curves (up to $g = 3$):

$$\begin{Bmatrix} \text{proper smooth genus} \\ g \text{ curves over } k \end{Bmatrix} \longrightarrow \begin{Bmatrix} \text{princ. pol. AVs of} \\ \text{dim. } g \text{ over } k \end{Bmatrix}$$
$$C \longmapsto \mathrm{Jac}(C) := \mathrm{Pic}^0_k(C)$$

## Other representations

- Jacobian of curves (up to $g = 3$):

$$\begin{Bmatrix} \text{proper smooth genus} \\ g \text{ curves over } k \end{Bmatrix} \longrightarrow \begin{Bmatrix} \text{princ. pol. AVs of} \\ \text{dim. } g \text{ over } k \end{Bmatrix}$$

$$C \longmapsto \mathrm{Jac}(C) := \mathrm{Pic}^0_k(C)$$

Also:

## Other representations

- Jacobian of curves (up to $g = 3$):

$$\left\{\begin{matrix}\text{proper smooth genus}\\ g \text{ curves over } k\end{matrix}\right\} \longrightarrow \left\{\begin{matrix}\text{princ. pol. AVs of}\\ \text{dim. } g \text{ over } k\end{matrix}\right\}$$
$$C \longmapsto \mathrm{Jac}(C) := \mathrm{Pic}^0_k(C)$$

Also:

- Prym varieties (up to $g = 5$)

## Other representations

- Jacobian of curves (up to $g = 3$):

$$\begin{Bmatrix} \text{proper smooth genus} \\ g \text{ curves over } k \end{Bmatrix} \longrightarrow \begin{Bmatrix} \text{princ. pol. AVs of} \\ \text{dim. } g \text{ over } k \end{Bmatrix}$$

$$C \longmapsto \mathrm{Jac}(C) := \mathrm{Pic}^0_k(C)$$

Also:

- Prym varieties (up to $g = 5$)
- Kummer varieties (more compact representation)

## Over $k = \mathbb{C}$

- An abelian variety over $\mathbb{C}$ of dimension $g$ is a complex torus.

$$A(\mathbb{C}) \simeq {}^{\mathbb{C}^g}\!\diagup_{L}$$

where $L$ is a lattice, that is, a free sub-$\mathbb{Z}$-module of $\mathbb{C}^g$ of rank $2g$.

# Over $k = \mathbb{C}$

- An abelian variety over $\mathbb{C}$ of dimension $g$ is a complex torus.

$$A(\mathbb{C}) \simeq {}^{\mathbb{C}^g}\!\big/_{L}$$

  where $L$ is a lattice, that is, a free sub-$\mathbb{Z}$-module of $\mathbb{C}^g$ of rank $2g$.
- Tori (coming from AVs) admit a Riemann form.

# Over $k = \mathbb{C}$

- An abelian variety over $\mathbb{C}$ of dimension $g$ is a complex torus.

$$A(\mathbb{C}) \simeq {}^{\mathbb{C}^g}\!\diagup_{L}$$

  where $L$ is a lattice, that is, a free sub-$\mathbb{Z}$-module of $\mathbb{C}^g$ of rank $2g$.
- Tori (coming from AVs) admit a Riemann form.
- We have an equivalence of categories:

$$\{\text{abelian varieties } /\mathbb{C}\} \longleftrightarrow \left\{ \begin{matrix} \mathbb{C}^g/L \text{ with } L \simeq \mathbb{Z}^{2g} \text{ with} \\ \text{eq.cl. of Riemann form} \end{matrix} \right\}$$

# Over $\mathbb{F}_q$

- In $\mathrm{char}(k) = p$ such an equivalence cannot hold.
- There are supersingular elliptic curves with quaternionic endomorphism algebra.
- In particular over $\mathbb{F}_q$, we need to restrict ourselves to sub-categories.
- There are various functors:
  - Deligne : ordinary AVs over any $\mathbb{F}_q$
  - Centeleghe-Stix : AVs with no real primes over prime fields $\mathbb{F}_p$
  - "Serre": AVs isogenous to power of "some" elliptic curves.

# My research: compute AVs up-to-isomorphism

- Fix an **ordinary squarefree** $q$-Weil polynomial $h$ :

# My research: compute AVs up-to-isomorphism

- Fix an **ordinary squarefree** $q$-Weil polynomial $h$ :
- by Honda-Tate theory $\rightsquigarrow$ an isogeny class $\mathscr{C}_h$.

# My research: compute AVs up-to-isomorphism

- Fix an **ordinary squarefree** $q$-Weil polynomial $h$ :
- by Honda-Tate theory $\rightsquigarrow$ an isogeny class $\mathscr{C}_h$.
- Put $K := \mathbb{Q}[x]/(h) = \mathbb{Q}[F]$.

## My research: compute AVs up-to-isomorphism

- Fix an **ordinary squarefree** $q$-Weil polynomial $h$ :
- by Honda-Tate theory $\rightsquigarrow$ an isogeny class $\mathscr{C}_h$.
- Put $K := \mathbb{Q}[x]/(h) = \mathbb{Q}[F]$.
- Deligne's equivalence induces:

Theorem (M.)

$$\left\{ \textit{abelian varieties over } \mathbb{F}_q \textit{ in } \mathscr{C}_h \right\}_{/\simeq}$$

## My research: compute AVs up-to-isomorphism

- Fix an **ordinary squarefree** $q$-Weil polynomial $h$ :
- by Honda-Tate theory $\rightsquigarrow$ an isogeny class $\mathscr{C}_h$.
- Put $K := \mathbb{Q}[x]/(h) = \mathbb{Q}[F]$.
- Deligne's equivalence induces:

**Theorem (M.)**

$$\left\{ \text{abelian varieties over } \mathbb{F}_q \text{ in } \mathscr{C}_h \right\}_{/\simeq}$$

$$\updownarrow$$

$$\left\{ \text{fractional ideals of } \mathbb{Z}[F, q/F] \subset K \right\}_{/\simeq}$$

# My research: compute AVs up-to-isomorphism

- Fix an **ordinary squarefree** $q$-Weil polynomial $h$ :
- by Honda-Tate theory $\rightsquigarrow$ an isogeny class $\mathscr{C}_h$.
- Put $K := \mathbb{Q}[x]/(h) = \mathbb{Q}[F]$.
- Deligne's equivalence induces:

**Theorem (M.)**

$$\left\{ \textit{abelian varieties over } \mathbb{F}_q \textit{ in } \mathscr{C}_h \right\}_{/\simeq}$$

$$\updownarrow$$

$$\left\{ \textit{fractional ideals of } \mathbb{Z}[F, q/F] \subset K \right\}_{/\simeq} \quad =: \mathrm{ICM}(\mathbb{Z}[F, q/F])$$

*ideal class monoid*

# Example

- Let $h(x) = x^8 - 5x^7 + 13x^6 - 25x^5 + 44x^4 - 75x^3 + 117x^2 - 135x + 81$.

# Example

- Let $h(x) = x^8 - 5x^7 + 13x^6 - 25x^5 + 44x^4 - 75x^3 + 117x^2 - 135x + 81$.
- $\rightsquigarrow$ isogeny class of an simple ordinary abelian varieties over $\mathbb{F}_3$ of dimension 4.

## Example

- Let $h(x) = x^8 - 5x^7 + 13x^6 - 25x^5 + 44x^4 - 75x^3 + 117x^2 - 135x + 81$.
- $\rightsquigarrow$ isogeny class of an simple ordinary abelian varieties over $\mathbb{F}_3$ of dimension 4.
- Let $F$ be a root of $h(x)$ and put $R := \mathbb{Z}[F, 3/F] \subset \mathbb{Q}(F)$.
- $\#\mathrm{ICM}(R) = 18 \rightsquigarrow$ 18 isom. classes of AV in the isogeny class.

# Example

Concretely:

$$
\begin{aligned}
I_1 =& 2645633792595191\mathbb{Z} \oplus (F + 836920075614551)\mathbb{Z} \oplus (F^2 + 1474295643839839)\mathbb{Z} \oplus \\
&\oplus (F^3 + 1372829830503387)\mathbb{Z} \oplus (F^4 + 1072904687510)\mathbb{Z} \oplus \\
&\oplus \frac{1}{3}(F^5 + F^4 + F^3 + 2F^2 + 2F + 6704806986143610)\mathbb{Z} \oplus \\
&\oplus \frac{1}{9}(F^6 + F^5 + F^4 + 8F^3 + 2F^2 + 2991665243621169)\mathbb{Z} \oplus \\
&\oplus \frac{1}{27}(F^7 + F^6 + F^5 + 17F^4 + 20F^3 + 9F^2 + 68015312518722201)\mathbb{Z} \\
I_7 =& 2\mathbb{Z} \oplus (F+1)\mathbb{Z} \oplus (F^2+1)\mathbb{Z} \oplus (F^3+1)\mathbb{Z} \oplus (F^4+1)\mathbb{Z} \oplus \frac{1}{3}(F^5 + F^4 + F^3 + 2F^2 + 2F + 3)\mathbb{Z} \oplus \\
&\oplus \frac{1}{36}(F^6 + F^5 + 10F^4 + 26F^3 + 2F^2 + 27F + 45)\mathbb{Z} \oplus \\
&\oplus \frac{1}{216}(F^7 + 4F^6 + 49F^5 + 200F^4 + 116F^3 + 105F^2 + 198F + 351)\mathbb{Z}
\end{aligned}
$$

# Example

Concretely:

$$
\begin{aligned}
I_1 =& 2645633792595191\mathbb{Z} \oplus (F + 836920075614551)\mathbb{Z} \oplus (F^2 + 1474295643839839)\mathbb{Z} \oplus \\
&\oplus (F^3 + 1372829830503387)\mathbb{Z} \oplus (F^4 + 1072904687510)\mathbb{Z} \oplus \\
&\oplus \frac{1}{3}(F^5 + F^4 + F^3 + 2F^2 + 2F + 6704806986143610)\mathbb{Z} \oplus \\
&\oplus \frac{1}{9}(F^6 + F^5 + F^4 + 8F^3 + 2F^2 + 2991665243621169)\mathbb{Z} \oplus \\
&\oplus \frac{1}{27}(F^7 + F^6 + F^5 + 17F^4 + 20F^3 + 9F^2 + 68015312518722201)\mathbb{Z} \\
I_7 =& 2\mathbb{Z} \oplus (F+1)\mathbb{Z} \oplus (F^2+1)\mathbb{Z} \oplus (F^3+1)\mathbb{Z} \oplus (F^4+1)\mathbb{Z} \oplus \frac{1}{3}(F^5 + F^4 + F^3 + 2F^2 + 2F + 3)\mathbb{Z} \oplus \\
&\oplus \frac{1}{36}(F^6 + F^5 + 10F^4 + 26F^3 + 2F^2 + 27F + 45)\mathbb{Z} \oplus \\
&\oplus \frac{1}{216}(F^7 + 4F^6 + 49F^5 + 200F^4 + 116F^3 + 105F^2 + 198F + 351)\mathbb{Z}
\end{aligned}
$$

$I_1$ is invertible in $R$, but $I_7$ is not invertible in $\mathrm{End}(I_7)$.

# Math Commercials : Computations

# Math Commercials : Computations

- With "Simons Collaboration on Arithmetic Geometry, Number Theory, and Computation"

## Math Commercials : Computations

- With “Simons Collaboration on Arithmetic Geometry, Number Theory, and Computation”
- computing on a server computer at the MIT.

# Math Commercials : Computations

- With "Simons Collaboration on Arithmetic Geometry, Number Theory, and Computation"
- computing on a server computer at the MIT.
- Data for $\sim 700.000$ isogeny classes for various $g$ and $\mathbb{F}_q$.

## Math Commercials : Computations

- With “Simons Collaboration on Arithmetic Geometry, Number Theory, and Computation”
- computing on a server computer at the MIT.
- Data for $\sim 700.000$ isogeny classes for various $g$ and $\mathbb{F}_q$.
- Some are pretty big: ($> 5$ millions isomorphism classes).

# Math Commercials : Computations

- With "Simons Collaboration on Arithmetic Geometry, Number Theory, and Computation"
- computing on a server computer at the MIT.
- Data for $\sim 700.000$ isogeny classes for various $g$ and $\mathbb{F}_q$.
- Some are pretty big: ($> 5$ millions isomorphism classes).
- Coming soon on the LMFDB (`https://www.lmfdb.org/Variety/Abelian/Fq/`)

# Non-Math Commercials: Where have I been ?

# Non-Math Commercials: Where have I been ?

- Bachelor: Universitá di Torino.
- Master: ALGANT (Universitá di Padova , Leiden University).
- PhD at Stockholm University (with Jonas Bergström).
- postdoc at the Max Planck Institute for Mathemamatics in Bonn.
- now: postdoc at Utrecht University (with Carel Faber).

## Non-Math Commercials: Where have I been ?

- Bachelor: Universitá di Torino.
- Master: ALGANT (Universitá di Padova , Leiden University).
- PhD at Stockholm University (with Jonas Bergström).
- postdoc at the Max Planck Institute for Mathemamatics in Bonn.
- now: postdoc at Utrecht University (with Carel Faber).

If you have any question about any of these places: ask away!

Thank you!