A Course on
Introduction to Arithmetic Dynamics

Mathematics Department
Khon Kaen University

Introduction to SageMath and Elliptic Curves

Chatchawan Panraksa

MUIC

May 26, 2023

Introduction to SageMath and Elliptic Curves

- Welcome to the introduction to computing elliptic curves using SageMath!
- SageMath is a powerful open-source mathematics software system that provides extensive functionality for working with mathematical objects, including elliptic curves.
- Elliptic curves are fundamental objects in mathematics and have various applications in cryptography, number theory, and algebraic geometry.
- In this series of slides, we will explore the capabilities of SageMath for performing computations on elliptic curves.
- By the end of this tutorial, you will gain hands-on experience in creating elliptic curve objects, performing basic operations, computing the order of curves, finding rational points, and exploring advanced topics such as cryptography applications and pairings.
- Let's dive in and begin our journey into the fascinating world of elliptic curves using SageMath!

Creating Elliptic Curve Objects in SageMath

- In SageMath, elliptic curve objects can be created using the 'EllipticCurve()' function.
- The general syntax for creating an elliptic curve object is:

  `E = EllipticCurve(K, [A, B])`

  (or `E = EllipticCurve(K, [a1, a2, a3, a4, a6])`)
  where `K` is the base field and `A` and `B` are the coefficients of the curve equation.
- For example, to create an elliptic curve over the rational field with equation $y^2 = x^3 + 2x + 3$, we can use:

  `E = EllipticCurve(QQ, [2, 3])`

  where `QQ` represents the rational field and the coefficients 2 and 3 are the values of a and b respectively.
- Once the elliptic curve object is created, various operations and computations can be performed on it.

Basic Operations with Elliptic Curves

- SageMath provides convenient methods for performing basic operations on elliptic curves.
- Point addition: Given two points P and Q on an elliptic curve E, we can compute the sum $P + Q$ using the 'add points' method:

  ```
  R = P + Q
  ```

  where `P`, `Q`, and `R` are the point objects representing the respective points on the curve.

Basic Operations with Elliptic Curves

- **Example:** Let's consider an elliptic curve $E : y^2 + y = x^3 - 7x + 6$ over the rational field $\mathbb{Q}$. Suppose we have two points $P = (2, 0)$ and $Q = (-1, 3)$ on the curve. To compute their sum $R = P + Q$, we can use the following code in SageMath:

```
E = EllipticCurve([0,0,1,-7,6])
P = E(2,0)
Q = E(-1,3)
R = P + Q
print(R)
```

  This will output the resulting point R on the curve E.

- Point doubling: To compute the doubling of a point P on the curve E, we can use the 'double()' method:

```
Q = 2*P
```

  where `P` and `Q` are the point objects representing the points on the curve.

Basic Operations with Elliptic Curves

- Scalar multiplication: We can perform scalar multiplication on a point P by an integer k using the 'scalar multiplication' method:

  ```
  Q = k*P
  ```

  where `P` and `Q` are the point objects representing the points on the curve, and `k` is the integer scalar.

Basic Operations with Elliptic Curves

- **Example** Let's consider an elliptic curve $E : y^2 = x^3 + 2x + 3$ over the rational field $\mathbb{Q}$. Suppose we have a point $P = (-1, 0)$ on the curve and we want to compute the scalar multiplication $Q = 3P$. Using SageMath, we can do the following:

```
E = EllipticCurve(QQ, [2, 3])
P = E(-1,0)
Q = 3*P
print(Q)
```

This will output the resulting point Q after performing scalar multiplication on P.

Computing the Order of an Elliptic Curve

- The order of an elliptic curve E is the number of rational points on the curve, including the point at infinity.
- SageMath provides the 'order()' method to compute the order of an elliptic curve.
- **Example** Consider the elliptic curve $F : y^2 = x^3 - 1$ over the rational field. We can compute its order in a similar manner:

```
F = EllipticCurve(QQ, [0,-1])
P= F(1,0)
order(P)
```

This will output the order of the curve F.

Computing the Order of a Point on an Elliptic Curve

- **Example 2:** Let's explore another elliptic curve $G : y^2 = x^3 - 2x$ over the rational field. Suppose we have a point R on G. We can compute its order using the following code:

```
G = EllipticCurve(QQ, [-2, 0])
R = G(0, 0)
order = R.order()
print(order)
```

This will output the order of the point R on the curve G.

Let's consider an elliptic curve $E : y^2 + y = x^3 - 7x + 6$ over the rational field $\mathbb{Q}$ again. Suppose we have two points $P = (2, 0)$ and $Q = (-1, 3)$ on the curve. To compute their sum $R = P + Q$, we can use the following code in SageMath:

```
E = EllipticCurve([0,0,1,-7,6])
P = E(2,0)
P.order()
```

Computing the Rank of an Elliptic Curve

- The rank of an elliptic curve E is a measure of the number of independent rational points on the curve.
- Computing the rank of an elliptic curve is a challenging problem in general.
- SageMath provides tools to estimate and compute the rank of an elliptic curve.
- The ‘rank()‘ function in SageMath can be used to compute the rank of an elliptic curve.
- The rank can give insight into the arithmetic properties and structure of the curve.

Computing the Rank: Example 1

- Let's consider the elliptic curve $E : y^2 = x^3 + 7x + 10$ over the rational field $\mathbb{Q}$.
- To compute the rank of E using SageMath, we can use the 'rank()' function.
- **Example:** Let's compute the rank of E and print the result:

```
E = EllipticCurve(QQ, [7, 10])
rank = E.rank()
print(rank)
```

This will output the rank of the curve E.

Computing the Rank: Example 2

- Consider another elliptic curve $F : y^2 = x^3 - 5$ over the rational field $\mathbb{Q}$.
- Let's compute the rank of F using the 'rank()' function in SageMath.
- **Example:** Let's compute the rank of F and print the result:

```
F = EllipticCurve(QQ, [0,-5])
rank = F.rank()
print(rank)
```

This will output the rank of the curve F.

Computing the Rank: Example 3

- Let's explore another elliptic curve $G : y^2 = x^3 - x$ over the rational field $\mathbb{Q}$.
- We can compute the rank of G using the 'rank()' function in SageMath.
- **Example:** Let's compute the rank of G and print the result:

```
G = EllipticCurve(QQ, [0,-1])
rank = G.rank()
print(rank)
```

This will output the rank of the curve G.

Computing the Torsion Subgroup

- SageMath provides the ‘torsion_subgroup()‘ function to compute the torsion subgroup of an elliptic curve.
- The syntax for using this function is:

  `torsion_subgroup(curve)`

  where `curve` is the elliptic curve object.
- This function returns a list of points representing the torsion subgroup of the curve, including the point at infinity.
- It allows us to explore the structure and properties of the torsion subgroup.

Example: Computing the Torsion Subgroup

- Let's consider the elliptic curve $E : y^2 = x^3 + 5x + 7$ over the rational field $\mathbb{Q}$.
- To compute the torsion subgroup of E using SageMath, we can use the 'torsion_subgroup()' function.
- **Example:** Let's compute the torsion subgroup of E and print the result:

```
E = EllipticCurve(QQ, [5, 7])
torsion_subgroup = E.torsion_subgroup()
print(torsion_subgroup)
```

This will output the list of points representing the torsion subgroup of the curve E.

Computing the Complete Torsion Subgroup

- To compute the complete list of the torsion subgroup of an elliptic curve, we can iterate through all possible points on the curve and check their orders.
- SageMath provides the 'torsion_points()' method to compute the complete torsion subgroup.
- The syntax for using this method is:

```
E = EllipticCurve(QQ, [a, b])
torsion_subgroup = E.torsion_points()
print(torsion_subgroup)
```

  where `E` is the elliptic curve object defined over the rational field $\mathbb{Q}$ with coefficients `a` and `b`.
- This method returns the complete list of torsion points on the curve, including the point at infinity.
- By examining the orders of the torsion points, we can determine the structure of the torsion subgroup.

Example: Computing the Complete Torsion Subgroup

- Let's consider the elliptic curve $E : y^2 = x^3 + 5x + 7$ over the rational field $\mathbb{Q}$.
- To compute the complete torsion subgroup of E using SageMath, we can use the 'torsion_points()' method.
- **Example:** Let's compute the complete torsion subgroup of E and print the result:

```
E = EllipticCurve(QQ, [5, 7])
torsion_subgroup = E.torsion_points()
print(torsion_subgroup)
```

This will output the complete list of torsion points on the curve E, including the point at infinity.

Generators of an Elliptic Curve

- Generators of an elliptic curve E are rational points on the curve that generate the entire group of points under the group law.
- A generator is also referred to as a *base point* or a *fundamental point*.
- The set of all points that can be obtained by adding the generator to itself or other points on the curve forms the *torsion subgroup*.
- Finding generators of an elliptic curve is essential for cryptographic applications, such as elliptic curve cryptography (ECC).
- SageMath provides methods to compute generators and explore the properties of elliptic curves.

Computing Generators

- SageMath offers the ‘gens()‘ method to compute the generators of an elliptic curve.
- The syntax for using this method is:

```
E = EllipticCurve(K, [a, b])
generators = E.gens()
print(generators)
```

  where `E` is the elliptic curve object defined over a field `K` with coefficients `a` and `b`.
- This method returns a list of generators for the elliptic curve E.
- The number of generators depends on the structure of the curve and its torsion subgroup.

Example: Computing Generators

- Let's consider the elliptic curve $E : y^2 = x^3 + 5x + 7$ over the rational field $\mathbb{Q}$.
- To compute the generators of E using SageMath, we can use the 'gens()' method.
- **Example:** Let's compute the generators of E and print the result:

```
E = EllipticCurve(QQ, [5, 7])
generators = E.gens()
print(generators)
```

This will output the list of generators for the elliptic curve E.

Elliptic Curves over Finite Fields

- Elliptic curves can also be defined over finite fields.
- An elliptic curve over a finite field $\mathbb{F}_q$ is defined by an equation of the form $E : y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_q$.
- The field $\mathbb{F}_q$ consists of q elements, where q is a prime power.
- The number of rational points on an elliptic curve over $\mathbb{F}_q$ is denoted as N_q and can vary.
- The Hasse's theorem gives an upper bound for the number of rational points as $|N_q - (q + 1)| \leq 2\sqrt{q}$.

Computing Rational Points over Finite Fields

- SageMath provides functions to compute the rational points on an elliptic curve over a finite field.
- The ‘rational_points()‘ method can be used to compute the rational points of an elliptic curve over a finite field.
- The syntax for using this method is:

```
E = EllipticCurve(GF(q), [a, b])
rational_points = E.rational_points()
print(rational_points)
```

  where `E` is the elliptic curve object defined over the finite field `GF(q)` with coefficients `a` and `b`.
- This method returns a list of rational points on the curve E over the finite field $\mathbb{F}_q$.

Example: Computing Rational Points over Finite Fields

- Let's consider the elliptic curve $E : y^2 = x^3 + 2x + 2$ over the finite field $\mathbb{F}_{13}$.
- To compute the rational points on E using SageMath, we can use the 'rational_points()' method.
- **Example:** Let's compute the rational points on E over $\mathbb{F}_{13}$ and print the result:

```
E = EllipticCurve(GF(13), [2, 2])
rational_points = E.rational_points()
print(rational_points)
```

This will output the list of rational points on the curve E over the finite field $\mathbb{F}_{13}$.

Example: Generators of Elliptic Curves over Finite Fields

- Let's consider the elliptic curve $E : y^2 = x^3 + 2x + 2$ over the finite field $\mathbb{F}_{13}$.
- To compute the generators of E using SageMath, we can use the 'gens()' method.
- **Example:** Let's compute the generators of E over $\mathbb{F}_{13}$:

```
E = EllipticCurve(GF(13), [2, 2])
generators = E.gens()
print(generators)
```

This will output the list of generators for the elliptic curve E over the finite field $\mathbb{F}_{13}$.