Lecture 28: Computing with Elliptic Curves (in PARI)

William Stein

Math 124 Harvard University **Fall 2001**

Contents

1 Initializing Elliptic Curves

We are concerned primarily with elliptic curves E given by an equation of the form

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

with a and b either rational numbers or elements of a finite field $\mathbb{Z}/p\mathbb{Z}$. If a and b are in $\mathbb{Q}$, we initialize E in PARI using the following command:

```
? E = ellinit([0,0,0,a,b]);
```

If you wish to view a and b as element of $\mathbb{Z}/p\mathbb{Z}$, initialize E as follows:

```
? E = ellinit([0,0,0,a,b]*Mod(1,p));
```

If $\Delta = -16(4a^3 + 27b^2) = 0$ then `ellinit` will complain; otherwise, `ellinit` returns a 19-component vector of information about E. You can access some of this information using the dot notation, as shown below.

```
? E = ellinit([0,0,0,1,1]);
? E.a4
%11 = 1
? E.a6
```

```
%12 = 1
? E.disc
%13 = -496
? E.j
%14 = 6912/31
? E5 = ellinit([0,0,0,1,1]*Mod(1,5));
? E5.disc
%15 = Mod(4, 5)
? E5.j
%16 = Mod(2, 5)
```

Here `E.j` is the *j-invariant* of E. It is equal to $\frac{2^8 3^3 a^3}{4a^3+27b^2}$, and has some remarkable properties that I probably won't tell you about.

Most elliptic curves functions in PARI take as their first argument the output of `ellinit`. For example, the function `ellisoncurve(E,P)` takes the output of `ellinit` as its first argument and a point `P=[x,y]`, and returns `1` if `P` lies on `E` and `0` otherwise.

```
? P = [0,1]
? ellisoncurve(E, P)
%17 = 1
? P5 = [0,1]*Mod(1,5)
? ellisoncurve(E5, P)
%18 = 1
```

2 Computing in The Group

The following functions implement some basic arithmetic in the group of points on an elliptic curve: `elladd`, `ellpow`, and `ellorder`. The `elladd` function simply adds together two points using the group law. Warning: PARI does *not* check that the two points are on the curve.

```
? P = [0,1]
%2 = [0, 1]
? elladd(E,P,P)
%3 = [1/4, -9/8]
? elladd(E,P,[1,0])    \\ nonsense, since [1,0] isn't even on E!!!
%4 = [0, -1]
? elladd(E5,P5,P5)
%12 = [Mod(4, 5), Mod(2, 5)]
? [1/4,-9/8]*Mod(1,5)
%13 = [Mod(4, 5), Mod(2, 5)]
```

The `ellpow` function computes $nP = P + P + \cdots + P$ (n summands).

```
? ellpow(E,P,2)
%5 = [1/4, -9/8]
```

```
? ellpow(E,P,3)
%6 = [72, 611]
? ellpow(E,P,15)
```

```
%7 = [26449452347718826171173662182327682047670541792/9466094804586385762312509661837302961354550401,
4660645813671121765025590267647300672252945873586541077711389394563791/920992883734992462745141522111225908861976098219465616585649245395649]
```

3 The Generating Function $L(E, s)$

Suppose E is an elliptic curve over $\mathbb{Q}$ defined by an equation $y^2 = x^3 + ax + b$. Then for every prime p that does not divide $\Delta = -16(4a^3 + 27b^2)$, the same equation defines an elliptic curve over the finite field $\mathbb{Z}/p\mathbb{Z}$. As you will discover in problem 3 of homework 9, it can be exciting to consider the package of numbers $\#E(\mathbb{Z}/p\mathbb{Z})$ of points on E over all finite fields. The function `ellap` computes

$$a_p(E) = p + 1 - \#E(\mathbb{Z}/p\mathbb{Z}).$$

```
? E = ellinit([0,0,0,1,1]);
? ellap(E,5)
%19 = -3          \\ this should be 5+1 - #points
? E5 = ellinit([0,0,0,1,1]*Mod(1,5));
? for(x=0,4, for(y=0,4, if(ellisoncurve(E5,[x,y]),print([x,y]))))
[0, 1]
[0, 4]
[2, 1]
[2, 4]
[3, 1]
[3, 4]
[4, 2]
[4, 3]
? 5+1 - 9          \\ 8 points above, plus the point at infinity
%22 = -3
```

There is a natural way to extend the definition of a_p to define integers a_n for every integer n. For example, if a_p and a_q are defined as above and p and q are distinct primes, then $a_{pq} = a_p a_q$. Today I won't tell you how to define the a_p when, e.g., $p \mid \Delta$. However, you can compute the numbers a_n quickly in PARI using the function `ellan`, which computes the first few a_n.

```
? ellan(E,15)
%24 = [1, 0, 0, 0, -3, 0, 3, 0, -3, 0, -2, 0, -4, 0, 0]
```

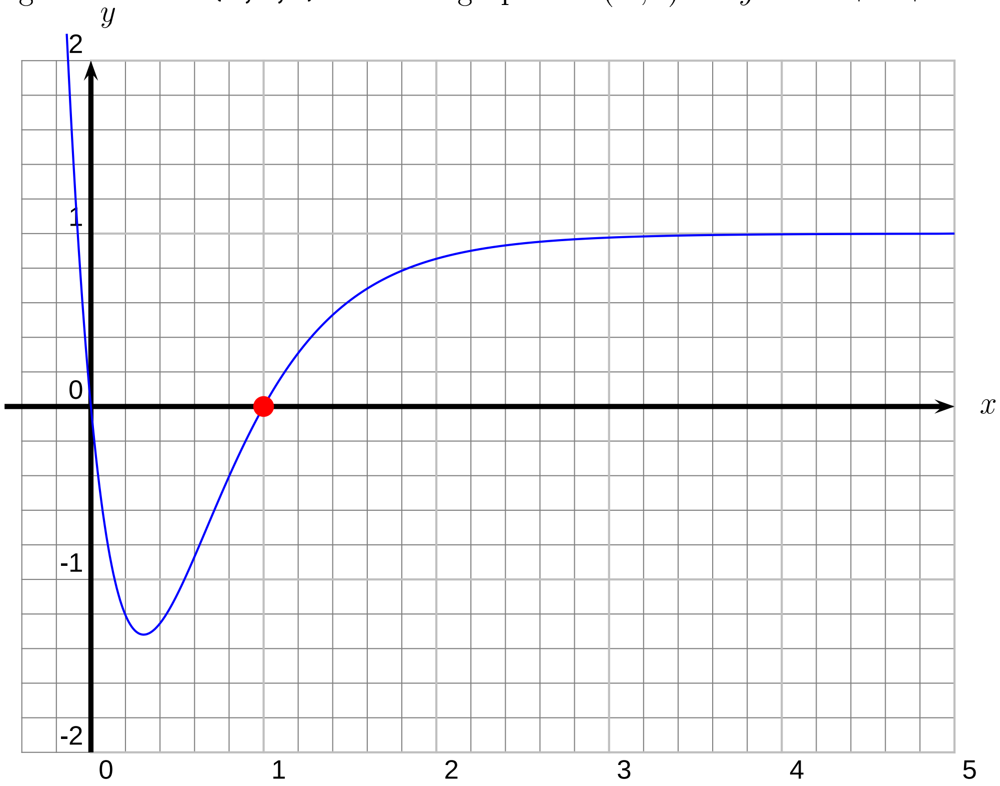
This output means that $a_1 = 1$, $a_2 = a_3 = a_4 = 0$, $a_5 = -3$, $a_6 = 0$, and so on.

When confronted by a mysterious list of numbers, it is a "reflex action" for a mathematician to package them together in a generating function, and see if anything neat happens. It turns out that for the above numbers, a good way to do this is as follows. Define

$$L(E, s) = \sum_{n=1}^{\infty} a_n n^{-s}.$$

This might remind you of Riemann's ζ-function, which is the function you get if you make the simplest generating function $\sum_{n=1}^{\infty} n^{-s}$ of this form.

Using `elllseries(E,s,1)` I drew a graph of $L(E,s)$ for $y^2 = x^3 + x + 1$.



That the value of $L(E,s)$ makes sense at $s = 1$, where the series above doesn't obviously converge, follows from the nontrivial fact that the function

$$f(z) = \sum_{n=1}^{\infty} a_n e^{2\pi i n z}$$

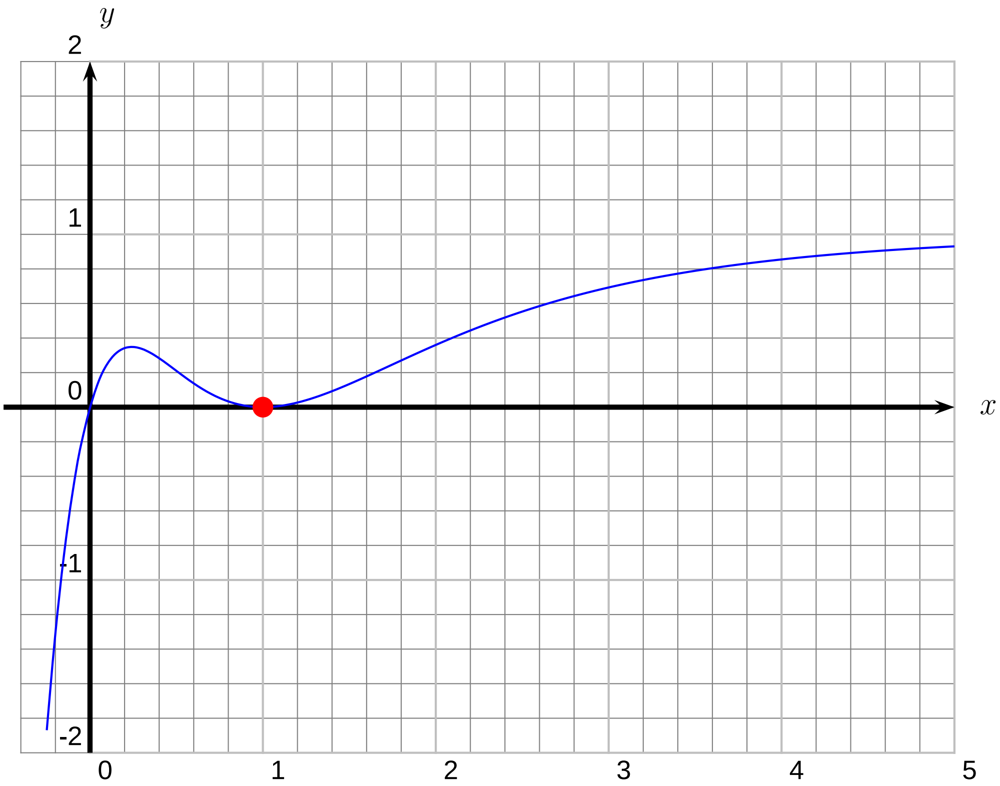
is a *modular form*. Also, keep your eyes on the dot; it plays a central roll in the Birch and Swinnerton-Dyer conjecture, which asserts that $L(E,1) = 0$ if and only if the group $E(\mathbb{Q})$ is infinite.

3.1 A Curve of Rank Two

Let E be the simplest rank 2 curve:

$$y^2 + y = x^3 + x^2 - 2x.$$

The discriminant is 389.

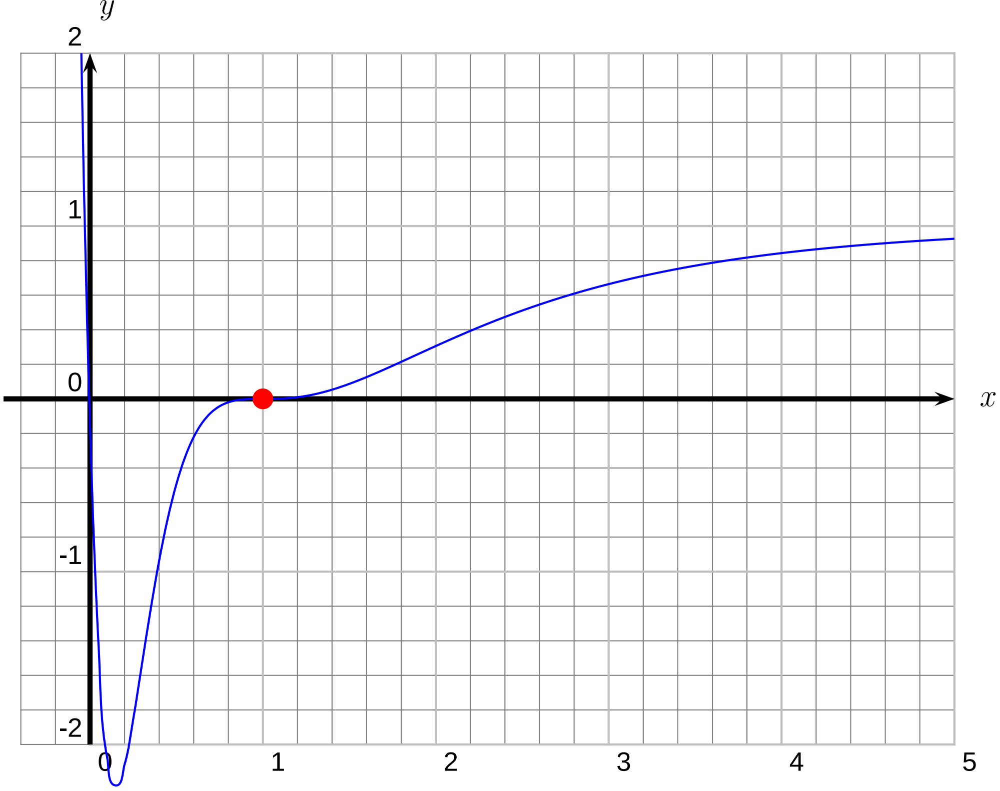


3.2 A Curve of Rank Three

Let E be the simplest rank 3 curve:

$$y^2 + y = x^3 - 7x + 6.$$

The discriminant is 5077.

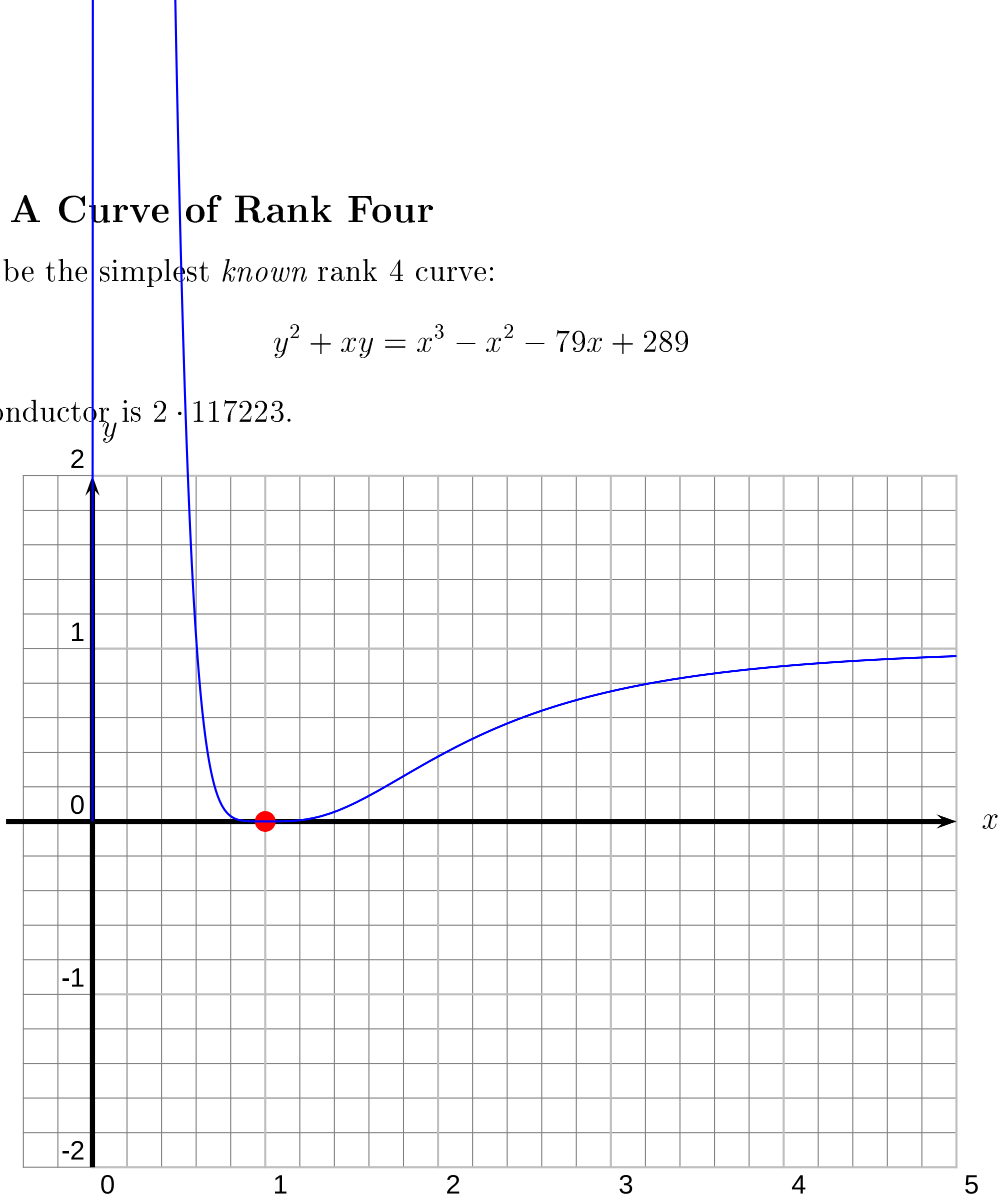


3.3 A Curve of Rank Four

Let E be the simplest *known* rank 4 curve:

$$y^2 + xy = x^3 - x^2 - 79x + 289$$

The conductor is $2 \cdot 117223$.



4 Other Functions and Programs

You can see a complete list of elliptic-curves functions by typing `?5`:

```
? ?5
elladd         ellak          ellan          ellap
ellbil         ellchangecurve ellchangepoint elleisnum
elleta         ellglobalred   ellheight      ellheightmatrix
ellinit        ellisoncurve   ellj           elllocalred
elllseries     ellorder       ellordinate    ellpointtoz
ellpow         ellrootno      ellsigma       ellsub
elltaniyama    elltors        ellwp          ellzeta        ellztopoint
```

I have only described a small subset of these. To understand many of them, you must first learn how to view an elliptic curve as a "donut", that is, as quotient of the complex numbers by a *lattice*, and also as a quotient of the upper half plane.

There is a Maple package called APECS for computing with elliptic curves, which is more sophisticated than PARI in certain ways, especially in connection with algorithms that involve lots of commutative algebra. MAGMA also offers sophisticated features for computing with elliptic curves, which are built in to the standard distribution. I will give a demonstrations of MAGMA in the Basic Notions seminar at 3pm on Monday, December 3 in SC 507. There is also a C++ library called LiDIA that has libraries with some powerful elliptic curves features.