Computing With the Birch and Swinnerton-Dyer Conjecture

William Stein
University of Washington, Seattle

Sage Days 5: Clay Math Institute
(get Sage at `http://sagemath.org`)

Opening Remarks

1. Welcome to Sage Days 5: Computational Arithmetic Geometry!
   Organizers: Jim Carlson, David Harvey, Kiran Kedlaya, and William Stein
2. **MANY THANKS to the Clay Mathematics Institute for generously fully funding this workshop!**
3. Many talks will be fairly specialized, so please do not hesitate to work during talks.
4. **Schedule change:** I swapped my BSD and Sato-Tate talks so Barry Mazur's can come to the Sato-Tate talk.

Elliptic Curves

Elliptic curves are (projective nonsingular) curves given by:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

```
sage: E = EllipticCurve([1,2,3,4,5]); E
Elliptic Curve defined by
          y^2 + x*y + 3*y = x^3 + 2*x^2 + 4*x + 5
sage: E.cremona_label()
'10351a1'

sage: EllipticCurve([1,2])
Elliptic Curve defined by y^2  = x^3 + x + 2 ...
sage: E = EllipticCurve(1); E          # j-invariant 1
          y^2 + x*y  = x^3 + 36/1727*x + 1/1727
sage: EllipticCurve('389a1')
          y^2 + y = x^3 + x^2 - 2*x
```

More Elliptic Curves: Cremona and Stein-Watkins

```
sage: P = SteinWatkinsPrimeData(0)

sage: C = P.next(); C.curves
[[[1, -1, 1, -1, 0], '[1]', '1', '4'],
 [[1, -1, 1, -6, -4], '[2]', '1', '2x'],
 [[1, -1, 1, -1, -14], '(4)', '1', '4'],
 [[1, -1, 1, -91, -310], '[1]', '1', '2']]

sage: cremona_optimal_curves([25..30])
<generator object at 0x2ad8cb0>

sage: list(cremona_optimal_curves([1..30]))
[Elliptic Curve defined by y^2 + y = x^3 - x^2 - 10*x - 20
 ...
 Elliptic Curve defined by y^2 + x*y + y = x^3 + x + 2
 over Rational Field]
```

Easy Invariants of Elliptic Curves

For an elliptic curve E over $\mathbb{Q}$, these invariants are easy to compute:

1. Discriminant
2. Conductor
3. Division polynomials
4. Period lattice; real volume Ω_E.
5. Root number

Example Computing Easy Invariants

```
sage: E = EllipticCurve([1..5])

sage: factor(E.discriminant())
-1 * 11 * 941

sage: factor(E.conductor())
11 * 941

sage: E.division_polynomial(3)
3*x^4 + 9*x^3 + 33*x^2 + 87*x + 35

sage: E.period_lattice()    # No control over precision!! (TODO)
(2.78074001376672977106319...

sage: E.omega()   # No control over precision!
2.78074001376672971063197...

sage: E.root_number()
-1
```

Elliptic Curves over Number Fields

For E over a number field, Sage computes:

1. **basic invariants** (as above),
2. 2-**descent** (Simon, Bradshaw)

Major gaps in functionality (that are all available in Magma):

1. No Tate's algorithm – to compute conductor, reduction – is not in Sage. Cremona's GPL'd Magma code will be ported (possible project for this week).
2. No L-series computation in any cases – could be built on top of Dokchitser's program. Sage also does *not* have fast computation of $\#E(\mathbb{F}_q)$ yet, via baby-step giant step, though it *almost* does (thanks to Martin Albrecht).
3. Height bounds over number fields. Again GPL'd Magma code of Cremona exists to do this, which needs to be ported.

Example over Number Fields

```
sage: K.<a> = NumberField(x^2 + 2)

sage: E = EllipticCurve([a, 3]); E
Elliptic Curve defined by y^2  = x^3 + a*x + 3 over
Number Field in a with defining polynomial x^2 + 2

sage: E.j_invariant()
-3359232/59177*a + 221184/59177

sage: E.simon_two_descent()
(0, -1, [])

sage: E.discriminant()
128*a - 3888
```

L-Series

Much functionality for L-series of elliptic curves over $\mathbb{Q}$:

1. $L(E,1)$ and $L'(E,1)$ with provable error bounds. (Stein)
2. $L^{(n)}(E,s)$ for any $n \geq 0$ and complex s to any requested precision, and the Taylor expansion about any point. (Dokchitser)
3. First m zeros of $L(E,s)$ in the critical strip to double precision. (Rubinstein)
4. p-adic L-series $\mathcal{L}_p(E,T)$. (Stein and Wuthrich)
5. Special values of symmetric powers $L(\mathrm{Sym}^{(n)}(E),s)$ of elliptic curve L-functions. (Watkins)

1. $L(E,1)$ and $L'(E,1)$ with provable bounds

Needed for my joint paper on "proving BSD for Cremona's book":

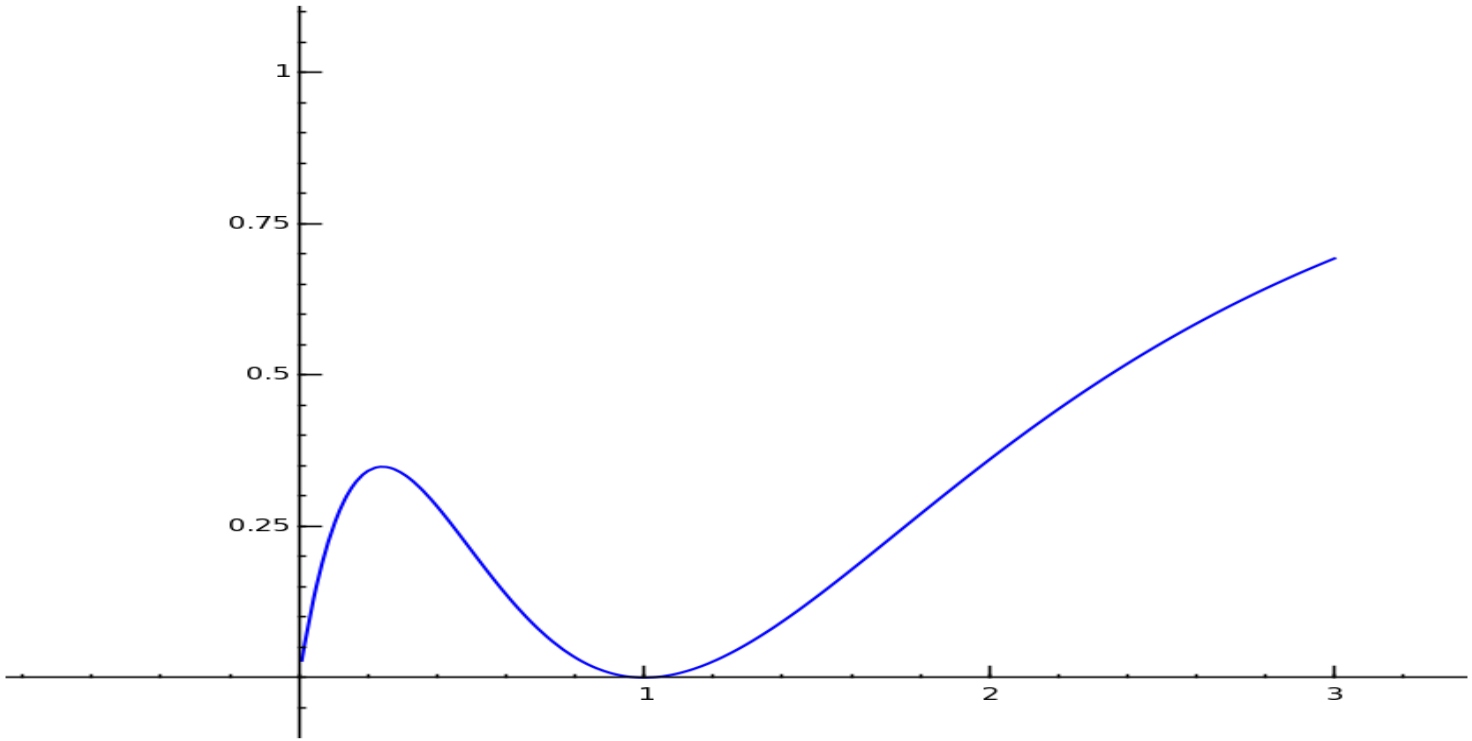
```
sage: E = EllipticCurve('37b')

sage: E.Lseries_at1(k=10)  # 10 terms
(0.725676956622683, 0.00003609675665441 75)

sage: E.Lseries_at1(k=100)  # 100 terms
(0.725681061936153, 1.52437502288743e-45)

sage: E = EllipticCurve('37a')

sage: E.Lseries_deriv_at1(k=10)  # 10 terms
(0.306000959182700, 0.00003609675665441 75)

sage: E.Lseries_deriv_at1(k=100)  # 100 terms
(0.305999773834879, 1.52437502288740e-45)
```

2. $L^{(n)}(E, s)$

```
sage: E = EllipticCurve('389a')
sage: L = E.Lseries_dokchitser()

sage: L(1)
-1.33174198778018e-19

sage: L(1+I)
-0.638409938588039 + 0.715495239204667*I

sage: L.derivative(1, 2)
1.51863300057685

sage: L.taylor_series(1)
-2.69129566562797e-23 + (1.52514901968783e-23)*z + ...

sage: L.taylor_series(I)
-0.764013101118315 - 9.4660116356710 8*I + (-19.889... + ...
```

Draw a plot of an L-series

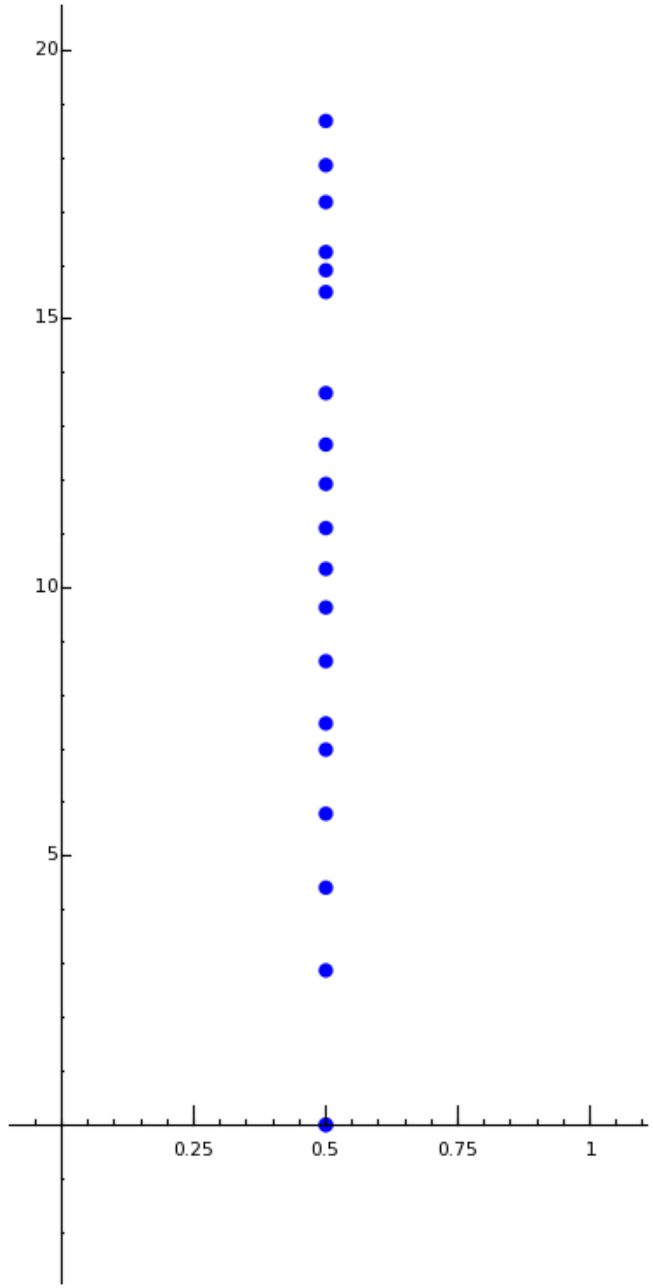
```
sage: E = EllipticCurve('389a')
sage: L = E.Lseries_dokchitser()

sage: show(plot(lambda x: abs(L(x)),0, 3),
           xmin=-0.5, ymin=0, dpi=150)
```



3. Zeros of $L(E, s)$ in the critical strip

```
sage: E = EllipticCurve('389a')

sage: time v = E.Lseries_zeros(20); v
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79340263,
 6.98596665, 7.47490750, 8.63320525, 9.63307880, 10.3514333,
11.1109355, 11.9335273, 12.6672137, 13.6248537, 15.5056185,
15.9115860, 16.2500699, 17.1798830, 17.8677033, 18.6909039]
CPU time: 0.01 s,  Wall time: 0.76 s

sage: show(list_plot([(1/2, y) for y in v],
                     pointsize=40), xmin=0, figsize=[4,8])
```

Zeros of $L(E, s)$... for 389a (rank 2 curve):



4. p-adic L-series $\mathcal{L}_p(E, T)$

```
sage: E = EllipticCurve('37a')
sage: L = E.padic_lseries(5)
sage: L.series(5)  # 5th approximation
```

$$O(5^7)+(1+4\cdot 5+2\cdot 5^2+5^3+O(5^4))\,T+(3+3\cdot 5^2+4\cdot 5^3+O(5^4))\,T^2+$$
$$(2+2\cdot 5+4\cdot 5^2+2\cdot 5^3+O(5^4))\,T^3+(4+3\cdot 5+5^2+3\cdot 5^3+O(5^4))\,T^4+\cdots$$

1. Every digit in output is correct (work of Koopa Koo, Pollack, Stein, Wuthrich).
2. Sage only – computing these is not included in Magma.
3. A wide range of curves treated (supersingular, ordinary, bad mult. reduction)
4. Not good for high precision – need Pollack-Stevens' p-adic overconvergent modular symbols algorithm.
5. These p-adic series are crucial to computational applications of Iwasawa theory to the BSD conjecture.

5. Example of a Symmetric Power L-function

```
sage: E = EllipticCurve('37a')
sage: sympow('-new_data 2')       # only do once
sage: E.Lseries_sympow(2, 16)
'2.492262044273650E+00'
```

Watkins' awesome Sympow also provides the world's best algorithm for computing modular degrees:

```
sage: E = EllipticCurve('5077a')  # rank 3
sage: time E.modular_degree()
1984
CPU time: 0.00 s,  Wall time: 0.01 s

sage: E = EllipticCurve([0, 0, 1, -79, 342])  # a rank 5 curve
sage: E.conductor()
19047851

sage: time E.modular_degree()           # amazing!!!
33108352
Time: CPU 0.03 s, Wall: 207.52 s
```

Computing the Mordell-Weil Group

Sage has excellent 2-descent code:

1. $E(\mathbb{Q})$ via 2-descent – Cremona's superb C++ programs (mwrank).
2. $E(\mathbb{Q})$ via algebraic 2-descent – Denis Simon's programs

Only Magma has these additional algorithms:

1. Heegner points method (Watkins)
2. 3-descent (Stoll)
3. 4-descent

Some Example Mordell-Weil Group Computations

```
sage: E = EllipticCurve([1,2,3,4,5])

sage: time E.gens()
[(1 : 2 : 1)]
CPU time: 0.01 s,  Wall time: 0.16 s

sage: E = EllipticCurve([12,2007])

sage: time E.gens()
[(448569/4096 : -300810003/262144 : 1)]
CPU time: 0.02 s,  Wall time: 0.20 s

sage: time E.simon_two_descent()
(1, 1, [(448569/4096 : 300810003/262144 : 1)])
CPU time: 0.05 s,  Wall time: 0.69 s
```

Regulators

1. Classical regulator of $E(\mathbb{Q})$ (TODO: add high precision).
2. Sage does not compute heights or regulators over number fields.
3. Sage is by far the best in the world at computing p-adic heights and regulators (Harvey, Stein, Bradshaw, Kedlaya, Mazur, Tate); important in p-analogues of the BSD conjecture.
4. Need to implement p-adic heights over number fields.

Examples computing classical and p-adic regulators

```
sage: E = EllipticCurve('389a')
sage: E.regulator()
0.15246017794314 4

sage: time E.padic_regulator(5, prec=10)
5^2 + 2*5^3 + 2*5^4 + 4*5^5 + 3*5^6 + 4*5^7 + O(5^8)
CPU time: 0.22 s,  Wall time: 0.25 s

sage: time E.padic_regulator(997, prec=10)
740*997^2 + 916*997^3 + 472*997^4 + 325*997^5 + 697*997^6
     + 642*997^7 + 68*997^8 + 860*997^9 + O(997^10)
CPU time: 0.44 s,  Wall time: 0.45 s

# INCREDIBLE ----------------------------\/!!!
sage: time E.padic_regulator(next_prime(10^5), prec=10)
42582*100003^2 + 35250*100003^3 + 12790*100003^4
     + 64078*100003^5 + 67283*100003^6 + 48411*100003^7
     + 7413*100003^8 + 22370*100003^9 + O(100003^10)
CPU time: 3.95 s,  Wall time: 4.50 s
```

The Birch and Swinnerton-Dyer Conjecture

Conjecture:
The rank r of $E(\mathbb{Q})$ equals $\mathrm{ord}_{s=1} L(E, s)$.

Conjecture: We have

$$\frac{L^{(r)}(E,1)}{r!} = \frac{\prod c_p \cdot \mathrm{Reg}_E \cdot \Omega_E \cdot \#Ш(E)}{\#E(\mathbb{Q})^2_{\mathrm{tor}}}.$$

And p-adic analogues with $L(E, s)$ replaced by $\mathcal{L}_p(E, T)$ and $\mathrm{Reg}(E)$ replaced by $\mathrm{Reg}_p(E)$.

There are major theorems about the p-adic versions (due to Kato, Skinner, Perrin-Riou, and others).

Compute Predicted Order of $Ш(E)$

In practice fairly straightforward to compute everything in the BSD conjecture except $Ш(E)$.

Compute:

1. Order of $Ш$ predicted by BSD Conjecture.
2. Order of $Ш$ predicted by the p-adic BSD conjecture; in many cases theorems imply that this order is correct up to a p-adic unit, assuming $\mathrm{Reg}_p(E) \neq 0$.

Examples of conjectural order computation

```
sage: E = EllipticCurve('389a')

sage: E.sha_an()
1.00000000000000

sage: time E.sha_an_padic(5, prec=3)  # Wuthrich, Stein
1 + O(5^2)
CPU time: 0.73 s,  Wall time: 0.90 s

sage: time E.sha_an_padic(23)
1 + O(23)
CPU time: 2.28 s,  Wall time: 2.59 s

sage: time E.sha_an_padic(97)
1 + O(97)
CPU time: 39.37 s,  Wall time: 45.24 s
```

Conclusion: $Ш(E/\mathbb{Q})[97] = 0$ (as predicted by BSD). Try that using descent!!

Kolyvagin and Kato's Theorems

Theorems (Kolyvagin, Kato): When E has (analytic) rank 0 or 1, explicit bounds on $Ш(E/\mathbb{Q})$ under certain hypothesis.

Sage can verify these hypothesis and compute bounds:

1. Heegner discriminants.
2. Index of Heegner subgroup in $E(K)$.
3. All primes p for which $\overline{\rho}_{E,p}$ is not surjective.
4. Whether $\overline{\rho}_{E,p}$ is irreducible.

Examples applying Kolyvagin's theorem

```
sage: E = EllipticCurve('37a')
sage: E.sha_an()
1
sage: E.analytic_rank()
1

sage: E.heegner_discriminants_list(10)
[-7, -11, -40, -47, -67, -71, -83, -84, -95, -104]
sage: E.heegner_index(-7)        # interval arithmetic
[0.99998569 .. 1.0000134]

sage: E.non_surjective()
[]
sage: E.shabound_kolyvagin()     # so only 2 divides Sha
([2], 1)
sage: E.two_selmer_shabound()    # bound on 2-rank of Sha
0
```

Thus $\#Ш(E/\mathbb{Q}) = 1$, hence the BSD conjecture is true for E.

Examples applying Kato's theorem

```
sage: E = EllipticCurve('37b')
sage: E.analytic_rank()
0
sage: E.non_surjective()
[(3, '3-torsion')]
sage: E.shabound_kato()
[2, 3]
sage: E.three_selmer_rank()          # calls magma
Traceback (most recent call last):
...
NotImplementedError:  Currently, only the case with irreducible
sage: E.two_selmer_shabound()
0
```

Thus Kato and 2-descent implies that only 3 could divide $\#Ш(E/\mathbb{Q})$. Moreover, Wuthrich's generalization of Kato's theorem implies that moreover $3 \nmid \#Ш(E/\mathbb{Q})$, so the BSD conjecture is true for E.

Thus BSD is true for the modular abelian variety $J_0(37)$.

Thanks. Questions?