Torsion Points on Elliptic Curves over Quartic Number Fields

Sheldon Kamienny and William Stein

August 25, 2011

Contents

1 Introduction

The *goal* of this paper is to prove the following conjecture:

Conjecture 1.1. *Suppose E is an elliptic curve over a quartic number field K. If a prime p divides $\#E(K)_{\mathrm{tor}}$ then $p \leq 17$.*

Something about [Jeon, Kim, Park]. Something about [Parent]

2 Geometric Arguments: $p = 19, 23, 29, 31, ...$

Kamienny's arguments that don't require using Section 3.

For 19 and 23 we get the result for fields in which at least one of 2, 3 doesn't remain prime. We can try dealing with 19 and 23 by looking (later) at equations for the modular curves if that's computable.

3 The Quartic Criterion

Let p be a prime and $\ell \neq p$ another prime. Let I_e be the annihilator in $\mathbb{T}_{\Gamma_1(p)}$ of the winding element $e = \{0, \infty\}$, so the winding quotient is $J_1(p)/IJ_1(p)$, which is an optimal quotient of $J_1(p)$, which may or may not be simple, but which does have rank 0 (by Kato's theorem).

Let $t_1 \in \mathbb{T}_{\Gamma_1(p)}$ be an element such that $t_1 I_e = 0$. We can construct examples of such elements as follows. Suppose $T \in \mathbb{T}_{\Gamma_1(p)}$. Factor $f = \prod f_i^{e_i}$ and let

$$J = \{j : (f/f_j^{e_j})(T)(\{0, \infty\}) = 0\}.$$

Then $t_1 = \prod_{j \in J} f_j^{e_j}(T)$ has the property that $t_1 I_e = 0$. The above calculation of J would be very expensive if done naively in general. Instead, we compute all the iterates $T^i(\{0, \infty\})$, so that we can then very efficiently compute $g(T)(\{0, \infty\})$ for *any* polynomial g.

If $\ell > 2$ let $t_2 = 1$. If $\ell = 2$, we take

$$t_2 = T_q - q\langle q \rangle - 1$$

for any choice of $q \neq p$.

Remark 3.1. In the definition of t_1, one could also use linear algebra to find a basis for the $\mathbb{Z}$-module of all possible t_1 that kill I, instead of just giving examples. This may turn out to be necessary. Likewise, Parent allows t_2 to more generally be any element that kills all μ_2 in $J_1(p)$, so if the above special examples of t_2 don't suffice, then we may have to find construct the space of all possible t_2.

Theorem 3.2 (Parent)**.** *Let d be an integer and p a prime number. Choose a prime number $\ell \neq p$ and two elements $t_1, t_2 \in \mathbb{T}$ as defined above, and let $t = t_1 t_2 \in \mathbb{T}$. For each partition $n_1 + n_2 + \cdots + n_m = d$ of d into m positive integers and $(m-1)$-tuple $v = (d_2, \ldots, d_m)$ with $1 < d_2, d_3, \ldots, d_m < p/2$ pairwise distinct, consider the following sequence*

$$B_{d,v} = \Big(tT_1, tT_2, \ldots, tT_{n_1}, \quad t\langle d_2 \rangle T_1, \ldots, t\langle d_2 \rangle T_{n_2}, \quad \ldots, \quad t\langle d_m \rangle T_1, \ldots, t\langle d_m \rangle T_{n_m}\Big)$$

of elements of $\mathbb{T} \otimes \mathbb{F}_\ell$. If $p > (1 + \ell^{d/2})^2$ and the elements of every sequence $B_{d,v}$ are linearly independent, then there is no elliptic curve over a degree d number field with a rational torsion point of order p.

We need an analogue of Proposition 1.12 of Parent but for degree 4 instead of degree 3. We take $d = 4$ and $\ell = 2$, so

$$(1 + \ell^{d/2})^2 = 25.$$

For $d = 4$, the partitions are:

$$(1,1,1,1), (1,1,2), (2,2), (1,3), (4).$$

Proposition 3.3. *Let $p > 25$ be a prime and consider Hecke operators T_n in the Hecke algebra $\mathbb{T} = \mathbb{T}_{\Gamma_1(p)} \otimes \mathbb{F}_2$ associated to $S_2(\Gamma_1(p); \mathbb{F}_2)$. Consider the following sequences of 4 elements of the Hecke algebra mod 2:*

1. *Partition 4=4:* (t, tT_2, tT_3, tT_4)
2. *Partition 4=1+3:* $(t, \quad t\langle d\rangle, t\langle d\rangle T_2, t\langle d\rangle T_3)$, *for $1 < d < p/2$.*
3. *Partition 4=2+2:* $(t, tT_2, \quad t\langle d\rangle, t\langle d\rangle T_2)$, *for $1 < d < p/2$.*
4. *Partition 4=1+1+2:* $(t, \quad t\langle d_1\rangle, \quad t\langle d_2\rangle, t\langle d_2\rangle T_2)$, *for $1 < d_1 \neq d_2 < p/2$.*
5. *Partition 4=1+1+1+1:* $(t, \quad t\langle d_1\rangle, \quad t\langle d_2\rangle, \quad t\langle d_3\rangle)$, *for $1 < d_1 \neq d_2 \neq d_3 < p/2$.*

If the entries in every single one of these sequences (for all choices of d_i) are linearly independent then there is no elliptic curve over a degree 4 number field with a rational point of order p.

To make the computations like the above efficient, the approach Parent takes is to explicitly represent the relevant Hecke algebra as univariant polynomial quotient ring over $\mathbb{Z}/3\mathbb{Z}$. Unfortunately, this only works in a few special cases, and is far less likely to work for $\ell = 2$. Already for $p = 29$ the approach in Parent seems to break down, since modulo 2 we do not have that T_2 is in the ring spanned by the diamond bracket operators.

An implementation in Sage of an algorithm based on the above proposition is given in the appendix (see Section 4). The next few subsections report on the results of running that algorithm.

3.1 $p = 29$

Running the code:

```
wstein@mod:~/comp/kamienny/may2010$ /usr/local/bin/sage go29.sage
Time: CPU 0.52 s, Wall: 0.52 s
J = []
partition 4=4...
partition 4=1+3...
```

```
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
(False, 'Fails with partition 4=1+1+1+1 and d1=2, d2=5, d3=12')
Time: CPU 7.03 s, Wall: 7.03 s
```

3.2 $p = 31$

Running the code:

```
wstein@mod:~/comp/kamienny/may2010$ /usr/local/bin/sage go31.sage
Time: CPU 0.52 s, Wall: 0.52 s
J = []
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
(False, 'Fails with partition 4=1+1+1+1 and d1=2, d2=5, d3=12')
Time: CPU 7.04 s, Wall: 7.04 s
```

3.3 $p = 37$

Running the code:

```
wstein@mod:~/comp/kamienny/may2010$ /usr/local/bin/sage go37.sage
Time: CPU 1.45 s, Wall: 1.45 s
J = [1]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
(True, 'All conditions are satified')
Time: CPU 44.96 s, Wall: 44.96 s
```

3.4 $p = 41$

Running the code:

```
sage: C = KamiennyCriterion(41)
sage: time C.verify_criterion()
J = []
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
```

```
partition 4=1+1+1+1...
(True, 'All conditions are satified')
Time: CPU 115.97 s, Wall: 139.65 s
```

3.5 $p = 43$

Running the code:

```
sage: time C = KamiennyCriterion(43); C
CPU times: user 1.01 s, sys: 0.13 s, total: 1.14 s
Wall time: 1.14 s
sage: time C.verify_criterion()
J = [0]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 84.17 s, sys: 0.01 s, total: 84.18 s
Wall time: 84.27 s
(True, 'All conditions are satified')
```

3.6 $p = 47$

```
sage: time C = KamiennyCriterion(47); C
CPU times: user 1.32 s, sys: 0.16 s, total: 1.48 s
Wall time: 1.48 s
sage: time C.verify_criterion()
J = []
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 178.90 s, sys: 0.01 s, total: 178.91 s
Wall time: 178.92 s
(True, 'All conditions are satified')
```

3.7 $p = 53$

```
sage: time C = KamiennyCriterion(53); C
CPU times: user 1.84 s, sys: 0.33 s, total: 2.17 s
Wall time: 2.18 s
sage: time C.verify_criterion()
J = [0]
partition 4=4...
```

```
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 407.35 s, sys: 0.12 s, total: 407.47 s
Wall time: 407.52 s
(True, 'All conditions are satified')
```

3.8 $p = 59$

```
sage: time C = KamiennyCriterion(59); C
CPU times: user 2.73 s, sys: 0.50 s, total: 3.23 s
Wall time: 3.23 s
sage: time C.verify_criterion()
J = []
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 951.67 s, sys: 0.36 s, total: 952.03 s
Wall time: 952.29 s
(True, 'All conditions are satified')
```

3.9 $p = 61$

```
sage: time C = KamiennyCriterion(61); C
CPU times: user 3.12 s, sys: 0.54 s, total: 3.66 s
Wall time: 3.66 s
sage: time C.verify_criterion()
J = [0, 1]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 1184.62 s, sys: 0.34 s, total: 1184.96 s
Wall time: 1185.82 s
(True, 'All conditions are satified')
```

3.10 $p = 67$

```
sage: time C = KamiennyCriterion(67); C
CPU times: user 4.33 s, sys: 0.87 s, total: 5.20 s
Wall time: 5.21 s
sage: time C.verify_criterion()
```

```
J = [1]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
^[^[^[CPU times: user 2356.57 s, sys: 1.08 s, total: 2357.65 s
Wall time: 2357.92 s
(True, 'All conditions are satified')
```

3.11 $p = 71$

```
sage: time C = KamiennyCriterion(71); C
CPU times: user 5.33 s, sys: 1.09 s, total: 6.42 s
Wall time: 6.41 s
sage: time C.verify_criterion()
pJ = []
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 3688.55 s, sys: 1.39 s, total: 3689.94 s
Wall time: 3690.35 s
(True, 'All conditions are satified')
```

3.12 $p = 73$

```
sage: time C = KamiennyCriterion(73); C
CPU times: user 6.28 s, sys: 1.18 s, total: 7.46 s
Wall time: 7.47 s
sage: time C.verify_criterion()
J = [3]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 4388.02 s, sys: 0.83 s, total: 4388.85 s
Wall time: 4389.08 s
(True, 'All conditions are satified')
```

3.13 $p = 79$

```
sage: time C = KamiennyCriterion(79); C
CPU times: user 8.03 s, sys: 1.45 s, total: 9.48 s
```

```
Wall time: 9.49 s
sage: time C.verify_criterion()
J = [0]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
n^[^[^[^[CPU times: user 8010.69 s, sys: 0.90 s, total: 8011.59 s
Wall time: 8015.41 s
(True, 'All conditions are satified')
```

3.14 $p = 83$

```
sage: time C = KamiennyCriterion(83); C
CPU times: user 9.94 s, sys: 1.77 s, total: 11.71 s
Wall time: 11.71 s
sage: time C.verify_criterion()
J = [0]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 11968.65 s, sys: 3.08 s, total: 11971.73 s
Wall time: 11979.33 s
(True, 'All conditions are satified')
```

3.15 $p = 89$

The following means we need to use an integral basis, which will take a long time.

```
sage: time C = KamiennyCriterion(89); C
CPU times: user 13.88 s, sys: 2.34 s, total: 16.22 s
Wall time: 16.22 s
sage: time C.verify_criterion()
J = [0]
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)

sage: time C = KamiennyCriterion(89, use_integral_structure=True); C
p = 89 with integral structure
Time: CPU 16.30 s, Wall: 16.30 s
J = [0]
partition 4=4...
partition 4=1+3...
```

```
partition 4=2+2...
partition 4=1+1+2...
partition 4=1+1+1+1...
Time: CPU 28860.55 s, Wall: 29003.75 s
```

3.16 $p = 97$

```
sage: time C = KamiennyCriterion(97); C
CPU times: user 20.14 s, sys: 3.79 s, total: 23.93 s
Wall time: 23.92 s
sage: time C.verify_criterion()
J = [0, 2]
partition 4=4...
partition 4=1+3...
partition 4=2+2...
        npartition 4=1+1+2...
partition 4=1+1+1+1...
CPU times: user 36558.55 s, sys: 239.50 s, total: 36798.05 s
Wall time: 36843.94 s
(True, 'All conditions are satified')
```

4 Appendix: The Code

For now, it is best to see

`http://wstein.org/home/wstein/comp/kamienny/code.sage`.

References

[CES03] B. Conrad, S. Edixhoven, and W. A. Stein, $J_1(p)$ *Has Connected Fibers*, Documenta Mathematica **8** (2003), 331–408.