George's example

February 1, 2022

1 George Turcas's example of Serre-Faltings

1.0.1 Define the base field:

[1]:
```
K.<i> = QuadraticField(-1)
```

1.0.2 Define the elliptic curve, find its conductor and set of bad primes:

[2]:
```
E = EllipticCurve([i+1,i-1,i+1,-5*i,2*i])
NE = E.conductor()
S = NE.prime_factors()
print(E)
print("Bad primes: {}".format(S))
```

```
Elliptic Curve defined by y^2 + (i+1)*x*y + (i+1)*y = x^3 + (i-1)*x^2 + (-5*i)*x
+ 2*i over Number Field in i with defining polynomial x^2 + 1 with i = 1*I
Bad primes: [Fractional ideal (i + 1), Fractional ideal (-5*i - 4)]
```

[3]:
```
%runfile C2C3S3.py
%runfile T0T1T2.py
```

1.1 Step 1

1.1.1 Find a set of primes T_0 such that the mod 2 representation is reducible if and only if the a_P for $P \in T_0$ are all even:

[4]:
```
get_T0(K,S)
```

[4]:
```
([], [], [])
```

1.1.2 $T_0 = \emptyset$! That is because K has no C_3 or S_3 extensions unramified outside S, so every elliptic curve over K with these bad primes must have 2-torsion:

[5]:
```
C3S3_extensions(K,S)
```

[5]:
```
[]
```

1.2 Step 2

1.2.1 Next we determine a set ofprimes T_2 for testing whether we have a large isogeny class (containing a curve with full 2-torsion, with trivial mod 2 representation) or a small class. Of course we know that the class is large since E has full 2-torsion:

[6]:
```
E.two_torsion_rank()
```

[6]:
```
2
```

1.2.2 Find T_2:

[7]:
```
T2_data = get_T2(K,S)
T2 = T2_data.values()
T2
```

[7]:
```
dict_values([Fractional ideal (2*i + 1), Fractional ideal (-3*i - 2), Fractional
ideal (i + 4), Fractional ideal (i - 4), Fractional ideal (2*i + 5), Fractional
ideal (-8*i - 7)])
```

1.2.3 According to Theorem 5.7 of [Cremona-Argaez], the class is large if and only if $F_P(1) \equiv 0 \pmod 4$ for all $P \in T_2$, where F_P is the Frobenius polynomial at P, so $F_P(1) = 1 - a_P + N(P)$. This is implemented as follows, where $t_1(P) = (1 - a_P + N(P))/2 \pmod 2$:

[8]:
```
BB = BlackBox_from_elliptic_curve(E)
t1 = BB_t1(BB)
[t1(P) for P in T2]
```

[8]:
```
[0, 0, 0, 0, 0, 0]
```

1.2.4 Since all are 0, the class is indeed large.

1.2.5 In more detail: the last number on each line should be a multiple of 4:

[9]:
```
norms = [P.norm() for P in T2]
traces = [E.reduction(P).trace_of_frobenius() for P in T2]
[(P, nP, aP, 1-aP+nP) for P,aP,nP in zip(T2,traces,norms)]
```

[9]:
```
[(Fractional ideal (2*i + 1), 5, -2, 8),
 (Fractional ideal (-3*i - 2), 13, 6, 8),
 (Fractional ideal (i + 4), 17, 2, 16),
 (Fractional ideal (i - 4), 17, 6, 12),
 (Fractional ideal (2*i + 5), 29, 10, 20),
 (Fractional ideal (-8*i - 7), 113, -14, 128)]
```

1.2.6 Note that we appear to be cheating here since we have obtained the traces from the elliptic curve, not from the Bianchi form. So we should check that for all the primes $P \in T_0 \cup T_2$ the eigenvalue of T_P on the Bianchi form is indeed equal to $a_P(E)$.

1.3 Step 3

1.3.1 To apply Livne's method we need a cubically independent set of primes for the space of quadratic extensions of K supported on S. Since $K(S,2) = \langle i, 1+i, 4+5i \rangle$ there are 7 of these. Since the dimension is only 3, in this case Livne is no better than the original Faltings–Serre which would require testing 8 primes P exhibiting all possible splitting behavious in these 7 extensions.

[10]:
```
vecs = []
plist = []
basis = [i,1+i,4+5*i]
np = 2**len(basis)
xK = polygen(K)
primes = primes_iter(K,1)
while len(plist)<np:
    P = next(primes)
    if P in S:
        continue
    vec = [lam(xK^2-d,P) for d in basis]
    if not vec in vecs:
        vecs.append(vec)
        plist.append(P)
        print("adding P={} with vector {}".format(P,vec))

[P.gens_reduced()[0] for P in plist]
```

```
adding P=Fractional ideal (-i - 2) with vector [1, 0, 0]
adding P=Fractional ideal (2*i + 1) with vector [1, 1, 0]
adding P=Fractional ideal (-3*i - 2) with vector [1, 0, 1]
adding P=Fractional ideal (i + 4) with vector [0, 1, 0]
adding P=Fractional ideal (i - 4) with vector [0, 1, 1]
adding P=Fractional ideal (-2*i + 5) with vector [1, 1, 1]
adding P=Fractional ideal (4*i + 5) with vector [0, 0, 0]
adding P=Fractional ideal (-8*i - 7) with vector [0, 0, 1]
```

[10]:
```
[-i - 2, 2*i + 1, -3*i - 2, i + 4, i - 4, -2*i + 5, 4*i + 5, -8*i - 7]
```

[ ]:
```

```