Bianchi mod 3

February 1, 2022

1 Example of mod 3 Galois representations over an imaginary quadratic field

1.1 This is an example where the $\mod 2$ representation is reducible and a 3-adic method should work better than 2-adic Faltings-Serre-Livne

1.1.1 Define the number field:

```
[1]: x = polygen(QQ)
     K.<i> = NumberField(x^2+1)
```

1.1.2 Define the elliptic curve and related quantities. This curve has LMFDB label 2.0.4.1-1312.1-b4

```
[2]: E = EllipticCurve([i + 1, i - 1, i + 1, -5*i, 2*i])
     show(E)
     S = E.conductor().prime_factors()
     show(S)
     S3 = S + K.primes_above(3)
     show(S3)
     f3 = E.division_polynomial(3)
     show(f3)
```

$$y^2 + (i+1)\,xy + (i+1)\,y = x^3 + (i-1)\,x^2 - 5ix + 2i$$

$$[(i+1)\,, (-5i-4)]$$

$$[(i+1)\,, (-5i-4)\,, (3)]$$

$$3x^4 + (6i-4)\,x^3 - 24ix^2 + 30ix - 10i + 1$$

1.1.3 Read in the files of code

[3]:
```
%cd /home/john/CremonaPacetti/code
%runfile S4.py
%runfile T0mod3.py
%runfile T0T1T2.py
```

```
/home/john/CremonaPacetti/code
```

1.1.4 Compute a set of test primes for the determinant character and use it to obtain the discriminant Δ such that $K(\sqrt{\Delta})$ is the fixed field of the kernel of the determinant character. We know that this is -3 (modulo squares) for an elliptic curve.

[4]:
```
T1, A, decoder = get_T1(K,S3)
print("T1:")
for P in T1:
    print("{}, norm {}".format(P,P.norm()))

BB = BlackBox_from_elliptic_curve(E)
decoder([0 if BB_det(BB)(P)%3==1 else 1 for P in T1])
```

```
T1:
Fractional ideal (2*i + 1), norm 5
Fractional ideal (-3*i - 2), norm 13
Fractional ideal (2*i + 3), norm 13
Fractional ideal (i + 4), norm 17
```

[4]:
```
3
```

1.1.5 That's equivalent to -3 mod squares since -1 is a square. We should get the same using the norms of the primes, since we know that for both the elliptic curve and the Bianchi newform (of weight 2) the determinant of the image of the Frobenius at P is $N(P) \pmod{3}$:

[5]:
```
decoder([0 if P.norm()%3==1 else 1 for P in T1])
```

[5]:
```
3
```

[6]:
```
S4quartics = S4_extensions(K,S3,D=-3)
D4quartics = D4_extensions(K,S3,d1=-3)
C4quartics = C4_extensions(K,S3,-3)
quartics = S4quartics+D4quartics+C4quartics
print("There are {} candidate quartics, of which {} are S4, {} are D4 and {}␣
 ↪are C4".format(len(quartics),len(S4quartics),␣
 ↪len(D4quartics),len(C4quartics)))
```

```
There are 95 candidate quartics, of which 79 are S4, 8 are D4 and 8 are C4
```

1.1.6 Compute T_0 and the associated 0-1 vectors:

[7]:
```
_, T0, vlist = get_T0_mod3(K,S3,quartics)
print("The test primes are")
for P in T0:
    print("P={} with norm {}".format(P,P.norm()))
```

```
The test primes are
P=Fractional ideal (i - 4) with norm 17
P=Fractional ideal (i - 6) with norm 37
P=Fractional ideal (4*i + 5) with norm 41
P=Fractional ideal (2*i + 7) with norm 53
P=Fractional ideal (5*i + 6) with norm 61
P=Fractional ideal (-3*i - 8) with norm 73
P=Fractional ideal (3*i - 8) with norm 73
P=Fractional ideal (-5*i + 8) with norm 89
P=Fractional ideal (-4*i + 9) with norm 97
P=Fractional ideal (4*i + 9) with norm 97
P=Fractional ideal (i - 10) with norm 101
P=Fractional ideal (10*i - 7) with norm 149
```

1.1.7 Compute the a_P for these primes from the elliptic curve:

[8]:
```
aplist = [BB_trace(BB)(P) for P in T0]
v0 = [0 if ap%3==0 else 1 for ap in aplist]
print("The traces for these primes are {}".format(aplist))
print("Test vector = {}".format(v0))
```

```
The traces for these primes are [6, 2, -6, 2, 14, -10, 6, 6, -14, -2, -10, -2]
Test vector = [0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1]
```

1.1.8 Test whether the test vector is in the list of test vectors. If it is then the representation is irreducible and we can get the associated quartic from the table:

[9]:
```
res = [i for i,vi in enumerate(vlist) if v0==vi]
if res:
    g3 = quartics[res[0]]
    print("The mod 3 representation is irreducible.")
    print("The splitting field of the projective representation is defined by␣
 ↪the quartic {}".format(g3))
    group = 'S4' if g3 in S4quartics else 'D4' if g3 in D4quartics else 'C4'
    print("The projective image is isomorphic to {}".format(group))
else:
    print("The representation is reducible")
```

```
The mod 3 representation is irreducible.
The splitting field of the projective representation is defined by the quartic
```

```
x^4 + (i - 1)*x^3 + 4*x + i - 1
The projective image is isomorphic to S4
```

1.1.9 Check that we have recovered the 3-division field of E:

[10]:

```
K.extension(f3,'t3').is_isomorphic(K.extension(g3,'u3'))
```

[10]:

```
True
```

1.1.10 Now we turn to the Bianchi modular form. This requires running my C++ code to get the Hecke eigenvalues a_P for P dividing $97, 113, \ldots, 197$:

[11]:

```
import subprocess

BIANCHI_PROGS_DIR = "/home/john/bianchi-progs"
MODULARITY_CHECKER = "/".join([BIANCHI_PROGS_DIR, "modularity_modp"])
os.chdir(BIANCHI_PROGS_DIR)

def ideal_gen_coeffs(I):
    return " ".join([str(c) for c in list(I.gens_reduced()[0])])

def apdata(E, P):
    ap = E.reduction(P).trace_of_frobenius()
    return " ".join([ideal_gen_coeffs(P), str(ap)])

def check_modularity(E, primes, verbose=False):
    K = E.base_ring()
    field = K.discriminant().squarefree_part().abs()
    ab = ideal_gen_coeffs(E.conductor())
    np = len(primes)
    input_string = " ".join([str(field), ab, "3", "1", str(np)] + [apdata(E,P)␣
 ↪for P in primes])
    if verbose:
        print("input string: {}".format(input_string))
    cmd = "echo {} | {}".format(input_string, MODULARITY_CHECKER)
    if verbose:
        print("command line: {}".format(cmd))
    pipe = subprocess.Popen(cmd, stdout=subprocess.PIPE, shell=True)
    if pipe.returncode:
        return None

    return pipe.stdout.read().decode().strip()
```

[12]:

```
check_modularity(E,T0)
```

[12]:

```
'b'
```

This means that there is a newform (with label ‘b’) of level E.conductor(), whose a_P are congruent mod 3 to those of E at the primes in T_0

Next we run the program moreap1 to compute the Hecke eigenvalues for both forms at this level at primes dividing $97,...,197”:

[13]:

```
#%cd /home/john/bianchi-progs
#
# Input parameters:
# 1 is the field (d=1, 2, 3, 7, 11 for Q(sqrt(-d)))
# 0 for verbosity level
# 36 4 for the level (36+4i)
# 97 113 ... 197: rational primes for which we compute the a_P for P|p
# 0 to signal no more primes
# 0 0 to signal no more levels
#
!echo "1 0 36 4 97 113 137 149 157 173 181 193 197 0 0 0" | /home/jec/
 ↪bianchi-progs/moreap1
```

```
Program moreap1: for given field and level, assumes that the newforms file
exists, and computes more individual Hecke eigenvalues.
---------------

Enter field: Verbose? Enter level (ideal label or generator): >>>> Level 1312.1
= (36+4i), norm = 1312 <<<<

2 newform(s) at level 1312.1 = (36+4i):
1:       basis = [];     aqlist = [ -1 -1 ];     aplist = [ 0 0 -2 -4 -6 2 -6 -4
10 0 -6 8 1 6 6 6 0 2 -10 -8 0 -8 -2 2 -16 2 -14 -2 8 -6 -14 -16 14 -2 -10 6 10
10 -14 18 -14 0 -16 22 2 18 -10 -20 2 0 14 10 18 -2 18 -18 -12 -30 18 12 0 -26
-10 -26 -6 -18 30 -10 16 22 22 -26 8 -14 -2 30 14 -18 -28 24 -12 -14 -2 -2 22
-26 -26 -6 -18 -18 -34 6 -4 2 -14 -42 22 2 -22 20 2 38 -38 -26 46 -14 -14 26 10
-8 -18 -16 6 2 30 -20 30 -12 14 -28 -44 22 0 18 48 42 2 10 14 14 -28 -6 -50 50
-14 50 -20 42 14 30 -6 -10 -20 26 8 34 -42 -54 6 34 -18 -10 -18 -6 -10 -48 38 56
-6 -18 -22 14 50 54 12 8 -38 -50 46 -34 -18 14 48 18 14 -60 -30 -6 30 -34 10 14
22 -18 -2 -62 54 -62 2 -30 -14 -18 -2 38 -4 0 52 14 -62 2 ]
Sign of F.E. = -1
Twisting prime lambda = 4+i, factor = 8
L/P ratio    = 0, cuspidal factor = 1
Integration matrix = [-9-18i,1;8-72i,3+2i], factor   = -1
2:       basis = [];     aqlist = [ 1 -1 ];      aplist = [ 0 2 -2 -2 6 -2 2 6 -2
10 6 2 1 -6 -6 -10 2 -2 14 -10 6 6 2 -14 -2 -18 -10 -2 10 -14 2 -10 -6 18 -2 -10
-10 14 -18 6 -10 18 -2 -6 6 -2 -10 10 -22 6 26 -14 2 -14 22 30 26 -26 -22 14 -30
6 10 -30 22 6 -22 -22 10 -10 26 -6 22 14 22 6 -26 30 -14 -34 22 10 -14 22 -2 26
26 -30 -6 -6 2 6 -30 -42 -34 -6 2 -14 14 18 -18 -2 42 18 -22 2 18 2 10 38 -2 2
-30 42 -22 -10 -34 -14 -2 10 38 26 -14 -42 -6 -2 46 38 -2 14 10 6 34 -6 18 34 -6
38 -34 -18 42 -30 10 6 10 30 -26 -10 2 -54 14 -50 -30 2 42 -2 -46 -10 6 -18 26
-30 2 42 -26 2 -26 58 58 6 46 -10 -22 42 2 30 -6 6 6 -50 30 -26 14 2 50 -18 22
-2 38 -6 -38 18 -62 6 -14 6 46 -30 2 -34 ]
Sign of F.E. = 1
Twisting prime lambda = 1, factor = -4
L/P ratio    = 1/4, cuspidal factor = 1
Integration matrix = [6+15i,1;-4+36i,2+i], factor   = -1
Making homspace and bases…done.
Enter a rational prime p (0 to finish):
ap for 4+9i: [ 2 -14 ]
ap for 9+4i: [ -16 -2 ]
Enter a rational prime p (0 to finish):
ap for 7+8i: [ -6 -14 ]
ap for 8+7i: [ -14 2 ]
Enter a rational prime p (0 to finish):
ap for 11+4i: [ 14 -6 ]
ap for 4+11i: [ -2 18 ]
Enter a rational prime p (0 to finish):
ap for 10+7i: [ -10 -2 ]
ap for 7+10i: [ 6 -10 ]
Enter a rational prime p (0 to finish):
ap for 11+6i: [ 10 -10 ]
ap for 6+11i: [ 10 14 ]
Enter a rational prime p (0 to finish):
ap for 2+13i: [ -14 -18 ]
ap for 13+2i: [ 18 6 ]
Enter a rational prime p (0 to finish):
ap for 9+10i: [ -14 -10 ]
ap for 10+9i: [ 0 18 ]
Enter a rational prime p (0 to finish):
ap for 7+12i: [ -16 -2 ]
ap for 12+7i: [ 22 -6 ]
Enter a rational prime p (0 to finish):
ap for 14+i: [ 2 6 ]
ap for 1+14i: [ 18 -2 ]
Enter a rational prime p (0 to finish):
Enter level (ideal label or generator):
```

1.1.11 From the above, taking the second eigenvalue of each pair (for form b, not form a) one can visually see that they agree with these:

[14]:

```
list(zip(T0,aplist))
```

[14]:

```
[(Fractional ideal (i - 4), 6),
 (Fractional ideal (i - 6), 2),
 (Fractional ideal (4*i + 5), -6),
 (Fractional ideal (2*i + 7), 2),
 (Fractional ideal (5*i + 6), 14),
 (Fractional ideal (-3*i - 8), -10),
 (Fractional ideal (3*i - 8), 6),
 (Fractional ideal (-5*i + 8), 6),
 (Fractional ideal (-4*i + 9), -14),
 (Fractional ideal (4*i + 9), -2),
 (Fractional ideal (i - 10), -10),
 (Fractional ideal (10*i - 7), -2)]
```

1.1.12 That means, given that we know that this Bianchi newform has level equal to the conductor of E, namely $(36+4i)$, that the projective $\mod 3$ representation attached to the Bainchi newform also has full image with the same splitting field as for E.

[ ]: